California State University, San Bernardino
CSUSB ScholarWorks

Theses Digitization Project | John M. Pfau Library

2004

Dynamic Report Generator

Liquan Xie

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd-project

 Part of the Databases and Information Systems Commons

Recommended Citation
Xie, Liquan, "Dynamic Report Generator" (2004). *Theses Digitization Project*. 2547.
https://scholarworks.lib.csusb.edu/etd-project/2547

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

DYNAMIC REPORT GENERATOR

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Li-Chuan Hsieh

December 2004

DYNAMIC REPORT GENERATOR

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Li-Chuan Hsieh

December 2004

Approved by:

[redacted]

Dr. Kay Zemoudeh, Chair, Computer Science — 6/15/04 Date

[redacted]

Dr. David Turner

[redacted]

Dr. Kerstin Voigt

ABSTRACT

This paper describes DRG (Dynamic Report Generator). This program can be used on the Internet to summarize the content of a database based on user criteria. DRG allows the user to choose a database, input the data, and generate a report based on user input, and print out the result in a new format. DRG can support several databases types. All data are transformed into a canonical XML format. DRG generates and stores the report in a new XML file. Users can generate a report from any database on the Internet. This is done through adding a Conversion Module for a new database without any changes to the report generator engine. This paper describes the design and analysis of the DRG program.

ACKNOWLEDGMENTS

The support of the National Science Foundation under award 9810708 is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Purpose

DRG is an Internet based dynamic report generator program; the user can get and generate reports over the Internet by using this program. Internet technology has improved a lot and this brings in more and more people to use the Internet. Internet users mostly chat, play games, and search for information. Therefore, most Internet programs are designed for these purposes. In these programs, the user only waits for the server's action, and then the user responds to these actions passively. The user does not have the ability to modify the data returned by the server. The user is passively getting information from the Internet. Furthermore, when the user gets information from the server, it is sent in a different format. If the user does not like the format, there is nothing she can do. Besides, if a user wants to generate a report by her, she must copy out the data and generates it by herself. This makes it more difficult for the user to use the data from the database.

DRG can access different databases. Because of the requirements for science and commerce purposes, different

databases (such as MYSQL, ACCESS, ORACLE, etc) exist have been invented. These databases have different designs, different functions, and different purposes. These databases make computer users more difficult to reach them because these databases have their own access program. For example, a user can use ACCESS to open an mdb file, but the user cannot use ACCESS to open an sql file. This situation also makes it more difficult for the computer programmer to design the program. Because the designer needs to build different programs to reach different databases. To solve this problem, we use XML. XML is a text-based database. It is easy to get and generate by program. Under the XML standard, computer software engineer can get, store, and generate data more easily. Because programmers do not need a complicated program to read and analyze the XML file, they just read the file and get the data. For this reason, DRG uses the XML as an internal format.

A user can use DRG remotely on any database and XML file. Because DRG is an Internet based program, the user does not need to go to a specific location or to use a specific computer to operate this program. Therefore, a user can use DRG anywhere in the world.

DRG is a platform independent program, because the DRG program is written in ASP.NET. It can work on Windows.

DRG has an independent module to accept the database. This module transforms the data of the database into a XML file. When the user wants to reach a new database, he/she does not need to rewrite the whole program. The user just changes this module to read the new database (see Figure 1).

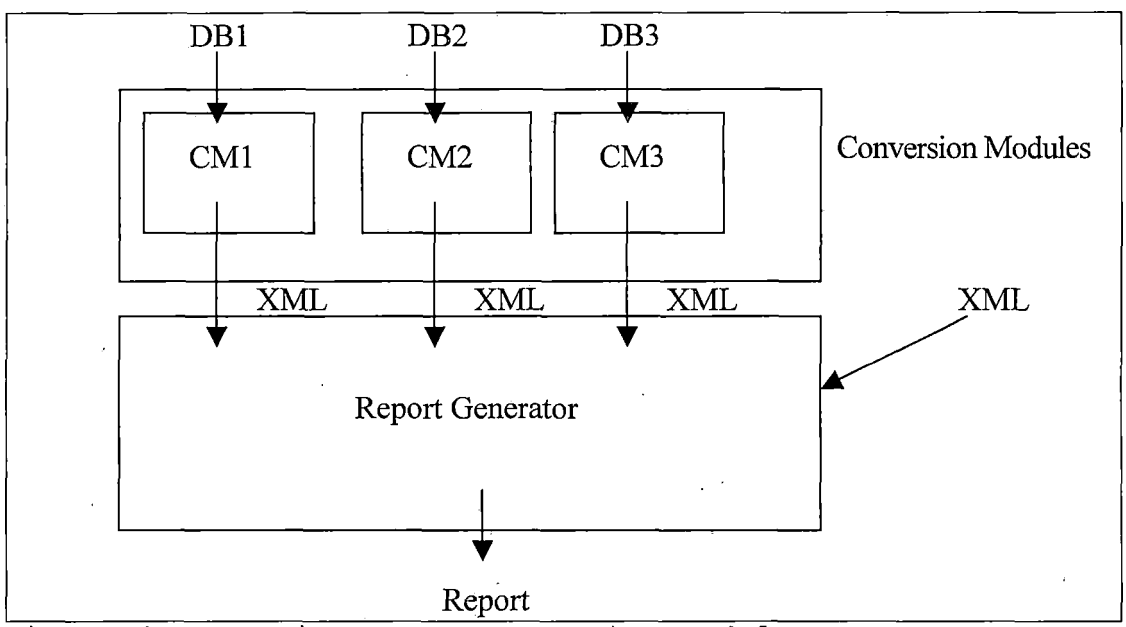

Figure 1. Dynamic Report Generator Module

DRG has the following aggregate functions for users to use: Average, Maximum, Minimum, Multiplication, and Count. Most report generators do not have these capabilities now, because these programs are not designed to generate the report. They only have sort, display, and

search. This means the current program just build a new layout for the report, not generate the contents of the report. To remove this defect, DRG adds these aggregate functions. The user can use these functions to generate a more effective report.

1.2 Description

DRG is built in ASP.NET. The algorithm is as follow,

a. The program accesses the URL address to get the file list.

b. The program downloads all files into a temporary directory at the client's machine.

c. The program generates the report.

d. The program prints out the result in a new layout.

Student_ID	Student_NAME	Student_Social
1	Sara	100-22-4444
2	Emily	237-35-4534
3	Wu	343-76-2454

Figure 2. Database Example (Student)

```
<Student>
 <Student_ID>1</Student_ID>
 <Student_NAME>Sara</Student_NAME>
 <Student_Social>100-22-4444</Student_Social>
</Student>
<Student>
 <Student_ID>2</Student_ID>
 <Student NAME>Emily</Student NAME>
  <Student_Social>237-35-4534</Student_Social>
</Student>
<Student>
  <Student_ID>3</Student_ID>
  <Student_NAME>Wu</Student_NAME>
  <Stuent_Social>343-76-2454</Student_Social>
</Student>
```

Figure 3. Extensible Markup Language Example (Student)

1.3 Significance

First, DRG can access many databases, and transform these databases into a XML document. In ASP.NET, VB can access a database and transform the whole database into a XML document directly.

Second, DRG can generate and sort the report content based on user's command. In ASP.NET, VB transfers XML document into a dataset. The program gets and generates data from this dataset based on the user's command.

1.4 Limitations

The biggest limitation of DRG is the server's policy. When DRG tries to access a URL, most of the time the program is denied access. In many servers' policy, the

server does not allow clients to reach files and databases for security reason. Therefore the program is stopped at the beginning most of the time. [6] [7]

1.5 Definitions

DRG - Dynamic Report Generator

VB - Microsoft Visual Basic Language

ASP.NET - Microsoft® ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web services. ASP.NET pages execute on the server and generate markup such as HTML, WML, or XML that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface.

ASP.NET pages and ASP.NET XML Web services file contain server-side logic (as opposed to client-side logic) written in Microsoft® Visual Basic .NET, Visual C# .NET, Visual C++. NET or any .NET framework-compatible language.

mdb - Microsoft Access Database.

XML - (Extensible Markup Language), XML is a subset of the Standard Generalized Markup language (SGML) defined

in ISO standard 8879:1986 that is designed to make it easy to interchange structured documents over the Internet. XML files always clearly mark the start and the end of logical parts (called elements) of a document. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web. It also defines how Internet Uniform Resource Locations (URL) identifies component parts of XML data streams.

SGML - (Standard Generalized Markup Language), SGML is a system for defining markup languages. Authors mark up their documents by representing structural, presentational, and semantic information alongside content. HTML is one example of a markup language.

CHAPTER TWO

REVIEW OF RELATED LITERATURE

2.1 JavaBean Dynamic Report Generator

JViewPro JavaBean (written entirely in Java) is a dynamic report generator and print preview control. Users can use JViewPro to build graphics and text-based reports or technical drawings, display the result in a scrolling and zooming viewer and then print it.

This program is a powerful commercial tool to make a dynamic report. It can create report layout, and changes the layout when new elements are added. Nevertheless, this program just uses the graphs and contents of a Homepage; it does not get the data from the database. Therefore, the utility of this program is limited. [10]

2.2 The Big Faceless Report Generator

The Big Faceless Report Generator is a Java component for converting the XML to PDF documents. This program uses JSP, ASP and similar technology, and the user can create dynamic PDF reports as quickly and easily as HTML.

This program just transforms an XML document into a PDF file. The program does not access databases (expect XML) and does not generate any report. [11]

2.3 Dynamic Grouping Report Generator

This is a commercial program. It provides a dynamic reporting environment with the flexibility and an entire set of features needed by the user. This program just selects the field and sorts them. This program works on the SQL database only. It sorts the records based on a selected field. [12]

2.4 Report Generator

This is a simple report generator that works for a school system. In this program, user inputs the student's name and information, and the program automatically builds the report sheet. This program does not access any database. [13]

2.5 User Report Generator

URG has gradually evolved with two aims. First, provide a tool that will allow competent users to create their own reports quickly. The program does this without any programming. Second, it gives the developers a tool that takes most of the tedium out of designing reports. Users can use this program to design the layout of the report. This program uses SQL as its internal format and it does not perform any database transformation. Therefore, this program's function is limited. [14]

CHAPTER THREE

METHODOLOGY

DRG is a tool for the Internet user to generate a report in any format. This program makes it easier for the Internet user to generate a report. The program is implemented in ASP.NET.

3.1 The Dynamic Report Generator Algorithm has Four Major Parts

3.1.1 Input Part

This program accesses the URL site to get the list of files.

a. The user inputs the URL Address.

b. The program accesses the URL address to get the list of files.

c. The program downloads all files into a temporary directory at the client's machine.

d. The program displays the list of databases.

e. The user chooses a database from the list.

3.1.2 Conversion Part

This program transforms the database to an XML format.

a. The program reads the data from the database

b. The program rewrites the data into an XML document.

3.1.3 Generate Part

This part generates the data in two ways.

a. Summary part: In this part, the user lays out his/her summary options. There are five options (Maximum, Minimum, Average, Multiplication, and Count) and one search bar.

b. Report part: In this part, the user makes one or more reports. The user lays out the report structure here. This part also has a search bar.

c. The user can do a pattern search in DRG. It means the user does not need to input the whole word; DRG can search by one character. For example, if the user wants to search for "John," she can input "John," "J*," "*h*," or "*n." Then the user can get the answer she wants. The search result of the Summary part is a number; it displays the number of records that matched the keyword. The search result in Report part is a table; this table displays all the matched record.

3.1.4. Output Part

This program collects the results of the Summary part and the Report part, and displays these results on the final page. All data transfer is done using XML.

3.2 Convert Module

Convert Module is designed to transform the database file into an XML document. Before DRG can generate the report, the database file must be transformed to an XML document by this module. This module is independent in DRG. The user can add a new module in this CM to access new database without changing the rest of the program. Therefore, DRG can easily access new database.

There are two functions in the CM. First function is input the data type into the main program. Second function is reads the database and transforms it to an XML document. It is easily for the user to add new CM by these two functions.

The next section (3.2.1) will discuss the CM function in ASP.NET program.

3.2.1 ASP.NET

In ASP.NET, each CM has two functions. First function is for add database's data type, and second function is for read data and write them into an XML document. When a

user wants to add a new database, her new CM must include these two parts.

In the first function, the user makes a simple function for stores the data type (like "mdb," "xml," etc). Just like this example:

```
Public Function FUNCTIONNAME ()
    dataarray.Add("***")
End Function
```

The user can put the data type in the place of the "*." This function stores the data type in to an ArrayList(). When DRG calls this function, this function sends the Arraylist() back to the main program.

```
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs)_ Handles MyBase.Load
FUNCTIONNAME()
□
□
```

Then the main program's getfilename() function uses this ArrayList() to search the database and get the file's name that matched the data type.

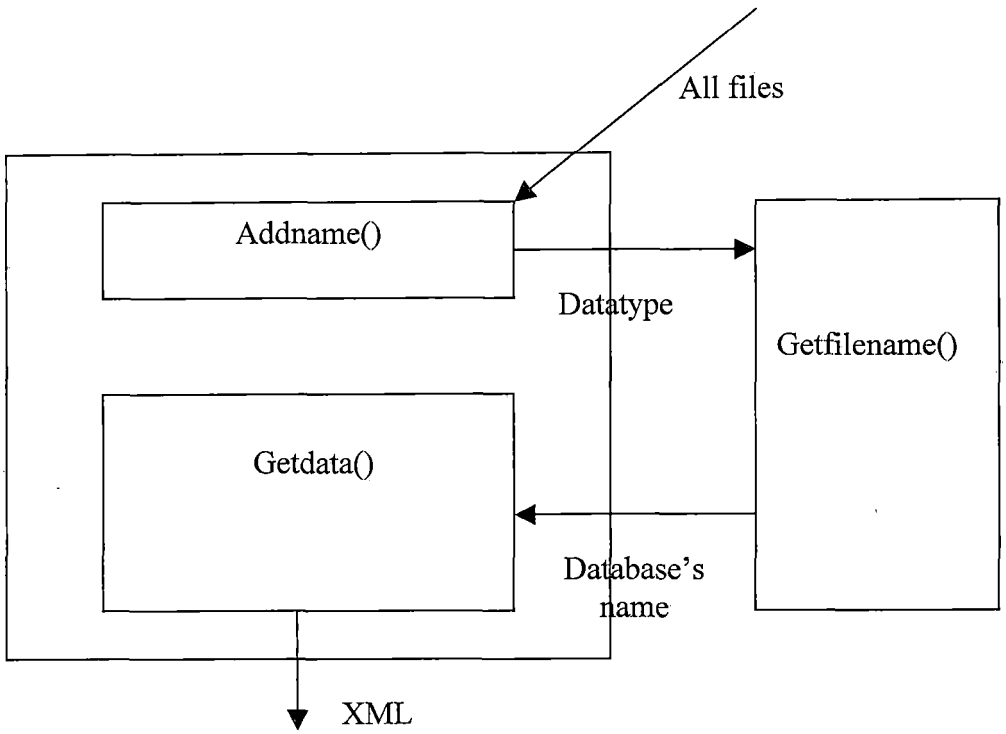

Figure 4. Convert Module Module

In the second function, the user needs to know how to access the database and put the data into a dataset by ASP.NET. For example, the user makes a function like this:

```
Public Function FUNCTIONNAME()
    Read a database and store the data into a
    dataset
    Write the data into an XML document
End Function
```

Then the user can use the ASP.NET command "dataset.WriteXml" to write whole dataset into a XML document. This "writexml" command can transform whole dataset into an XML document with schema and save this XML

document into a specific directory. When DRG calls this function, it sends back an XML document.

```
Dim title = datachoose.ToCharArray
(datachoose.LastIndexOf(".") + 1, 3)
Select Case title
    Case Is = "mdb"
        getaccessdata() 'Call
```

Example: This is a CM example for ACCESS database. The first function sends back the data type "mdb." The second function sends back an XML document.

```
'Add access data type
Public Function addname1()
    dataarray.Add("mdb")
    End Function
'Access database and transforms it into an XML document
Public Function getaccessdata()
    Dim title = datachoose.ToCharArray
    (datachoose.LastIndexOf(".") + 1, 3)
ListBox1.Items.Add(title)
Dim objCat
Dim objTables
Dim objTable
objCat = Server.CreateObject("ADOX.Catalog")
```

```
objCat.ActiveConnection = "provider=microsoft.jet.
oledb.4.0;" & "data source=" &_ Server.MapPath
(datachoose)
objTables = objCat.Tables
For Each objTable In objTables
   If objTable.Type = "TABLE" Then
      Dim connstr As String = "provider=microsoft.
      jet.oledb.4.0;data source=" &_ Server.MapPath
      (datachoose)
      Dim conn As OleDbConnection, adap As
      OleDbDataAdapter
      conn = New OleDbConnection(connstr)
      conn.Open()
      Dim sql = "select * from " & objTable.Name
      adap = New OleDbDataAdapter(sql, conn)
      adap.Fill(mydataset, objTable.Name)
      mydataset.WriteXml(Server.MapPath("temp.xml"),
      XmlWriteMode.WriteSchema)
      conn.Close()
End If
   Next
   objCat = Nothing
   datachoose = "temp.xml"
   Session("datachoose") = datachoose
```

```
        Session("datachoose2") = datachoose2
    End Function
Example: This is the example to show where the call is.
    Private Sub Form1_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load
    addname1() 'Call
    addname2() 'Call
    addname3() 'Call
    End Sub
    Private Sub Button1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button1.Click
    Dim title = datachoose.ToCharArray
    (datachoose.LastIndexOf(".") + 1, 3)
    Select Case title
      Case Is = "mdb"
        getaccessdata() 'Call
      Case Is = "xml"
        getxml()        'Call
      Case Is = "sql"
        getoracle()     'Call
    End Select
    Response.Redirect("choose3.aspx")
  End If
End Sub
```

In conclusion, when the user wants to add a new database, he/she has several steps to do.

a. The user makes the first function to add new database's data type.

   For example, the user wants to add a new data type "abc."

```
Public Function addabc()
    dataarray.Add("abc")
End Function
```

b. The user makes the second function to access the new database and transform it into an XML document.

```
Public Function getabc()
    Read a database and store the data into a dataset
    Write the data into an XML document
End Function
```

c. The user puts the calls in the main program. When the user adds the second function call in the main program. The user needs to add a new case condition.

```
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    addabc() 'Call addabc() function
End Sub
```

```
    Private Sub Button1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button1.Click
    Dim title = datachoose.ToCharArray(datachoose.
    LastIndexOf(".") + 1, 3)
    Select Case title
    Case Is = "abc" 'New Case, to check is it match the
    "abc"
      getabc()     'Call getabc() function
    End Select
      Response.Redirect("choose3.aspx")
    End If
End Sub
```

3.3 Designs and Interface

3.3.1 The Detail of ASP.NET Interfaces

3.3.1.1 Start Page. In this page, the user needs to input the URL address of the database. There are three textbox, one confirm button, and four titles in this page. The user inputs the URL address, username, and password in the textboxs, and presses the confirm button. The program would accesses the URL address, get the file's list, and downloads all files into a temporary directory at the client's machine.

The major point of this page is to access the URL address, get the file's list, and download the files. At first, the program will check the input text of the URL address to make sure it is not a blank. When the answer is positive, the program accesses this URL address and opens the connection with this URL address. Then the program create a arraylist() to store the file's list, and start downloads the files from this URL address. At the same time, the program makes a temporary directory at the client's machine, and uses the directory to store the files that downloads from the URL address. When the program success gets the file's list, it uses the "Session" command to transfer the arraylist() to the next page. The "Session" command is used to store the information when the user jumps between the pages in the application. Variables stored in the Session object are not discarded when the user change the pages (see Figure 5).

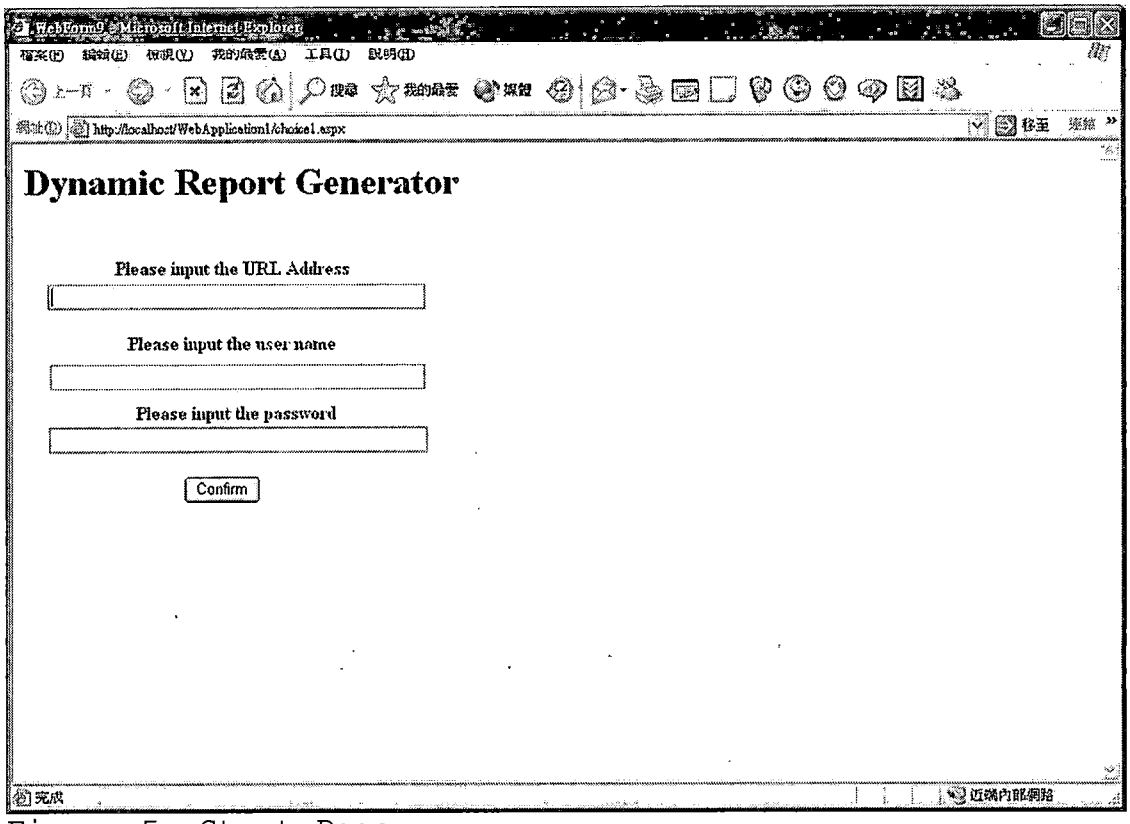

Figure 5. Start Page

3.3.1.2 Database Display Page. This page is designed to show all databases from the URL address that the user entered in the first page. The user can choose the database that the report is to be generated from, and these databases are store at the temporary directory. There is one pane and one confirm button in this page. This pane displays the list of the Access database, XML document, and other database's list. The user can choose one of these databases in the pane and presses the confirm button to go to next page. Under the first pane is the table of database lists. This table displays the details

of the database files (name, length, and access time) that came from the ULR address. When the user goes to the next page, the database's data is transformed to an intermediate XML document (see Figure 6).

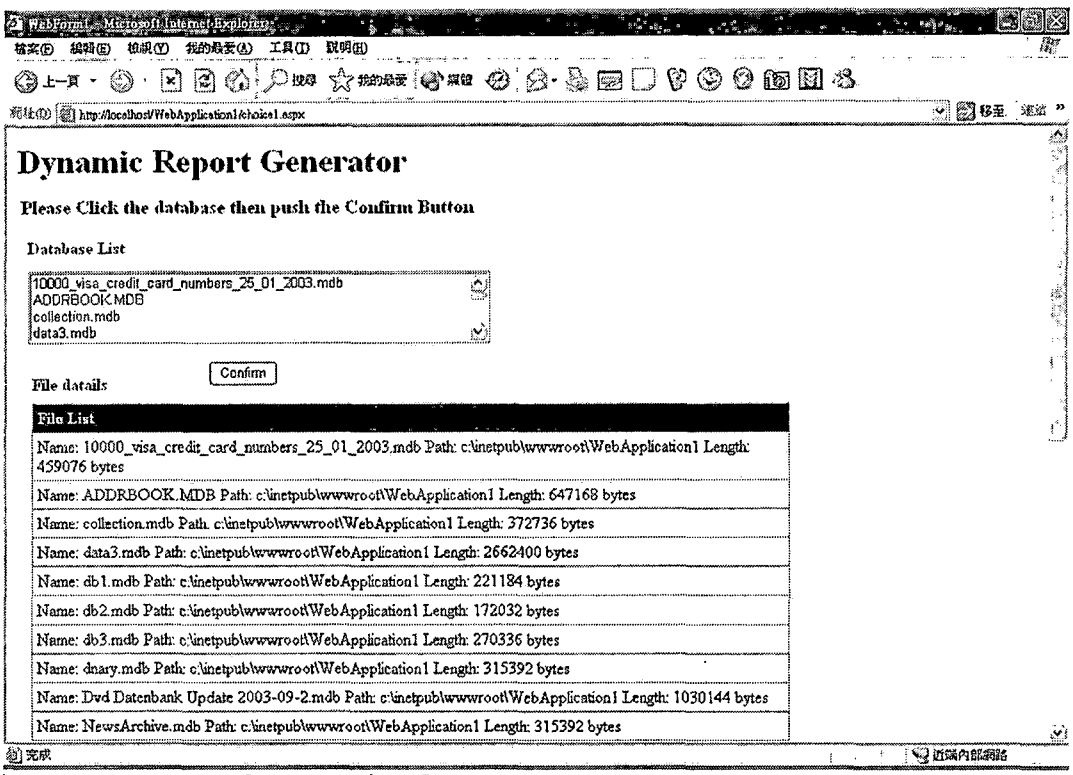

Figure 6. Database Display Page

3.3.1.3 Option Selection Page. The user chooses between the Summary and the Report option in this page. After the user chooses one, he/she also needs to confirm the selection and go to the option page (see Figure 7).

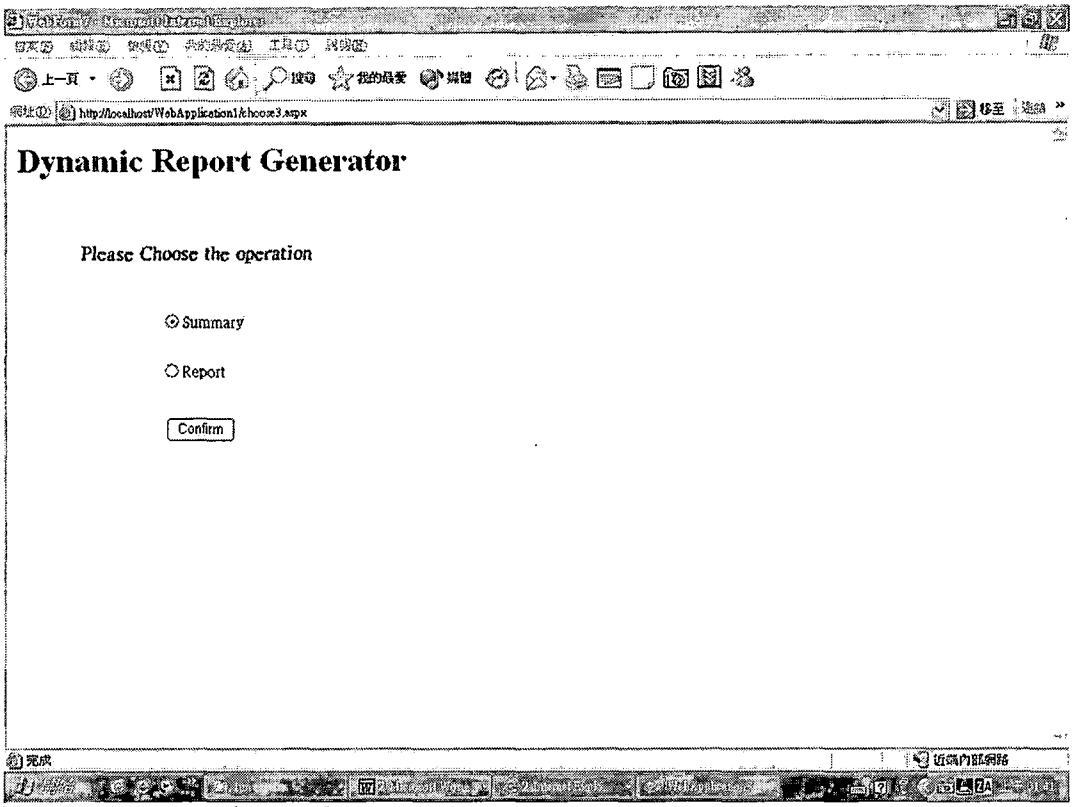

Figure 7. Option Choose Page

3.3.1.4 Summary Page. The DRG program generates a summary report in this page.

a. The user sees a table at the upper left corner of the page. It shows the database's detail. It includes each table's name and columns. The user can see the schema of the database here.

b. The right hand side of the page is the option part. There are four dropdownlist boxes and one text box here.

i. The first dropdownlist box is used for choosing the table. The user can choose any one of the tables of the database here.

ii. The second dropdownlist box is used for choosing columns. The user can choose any column of the table here.

iii. The third dropdownlist box is used for choosing an aggregate function. There are five options in this box. They are Average, Maximum, Minimum, Multiplication, and Count. When the user chooses Average, Maximum, Multiplication and Minimum, the user can continue to the next step. When the user chooses the "Count," she needs to choose the logical option and input the require number. For example, if a user wants to know how many student's grades is greater than 50, the user needs to choose the "Count" option, input the number 50, and choose the logical option ">." Then the user will get the answer. There are three logical options, they are ">," "<," and "=."

iv. Under these dropdownlist boxes, the user can see the button names "Check Data" (See Figures 8 and 10). This button shows the first ten rows of the table. User uses this button to check table's data.

c. Below the table list window, there are four buttons. These buttons are Delete, Clear, Save, and Exit.

i. When the user chooses the table, column, and option. The user can press the Save button to save the chosen selection. The user can see the chosen selection in the data list that under these buttons. If the data cannot be generated, the program shows an error message to remind the user.

ii. If the result does not satisfy the user, the user can press the Delete button to delete the last chosen selection or press the Clear button to delete all chosen selection.

iii. When the user has finished this page, the user can press the Exit button to leave this page and go to next page.

iv. The user can input a keyword in the textbox and press the Search button to search the database. These search results are displayed as numbers; they indicate how many records have matched the keyword in this data table. For example, if the user wants to search the "Grades" column in the "Student" table. She inputs the "5*" in the search textbox. "5*" is means start with "5." Then the search result she gets is number "4." This means there are four records are matched the "5*" in the "Grades" column.

d. Below the buttons, there is a textbox. The user can use this textbox to input the Header's name or title (see Figure 8).

3.3.1.5 Report Page. Most components of this page are the same as Summary page. The major difference between these two pages is that there is no option choice in Report page. The user needs to choose a table and a column, and then press the Save button. The result is displayed in a datalist. The result includes table's name and columns. When the user changes the table, the table's name will be changed dynamically and is displayed in the

datalist. The keyword search is different here from the summary page. The results of the keyword search are displayed completely in a table.

In the Summary part and the Report part, the program uses XML document command (DataSet.ReadXml) to import XML document into a dataset. DRG uses this dataset to find out the table's name and columns, and then inputs them into a dropdownlist box. When the user needs to get and generate a report from XML, DRG creates a dataset and transforms the whole XML data into this dataset. Then DRG uses this dataset to generate the report (see Figure 9).

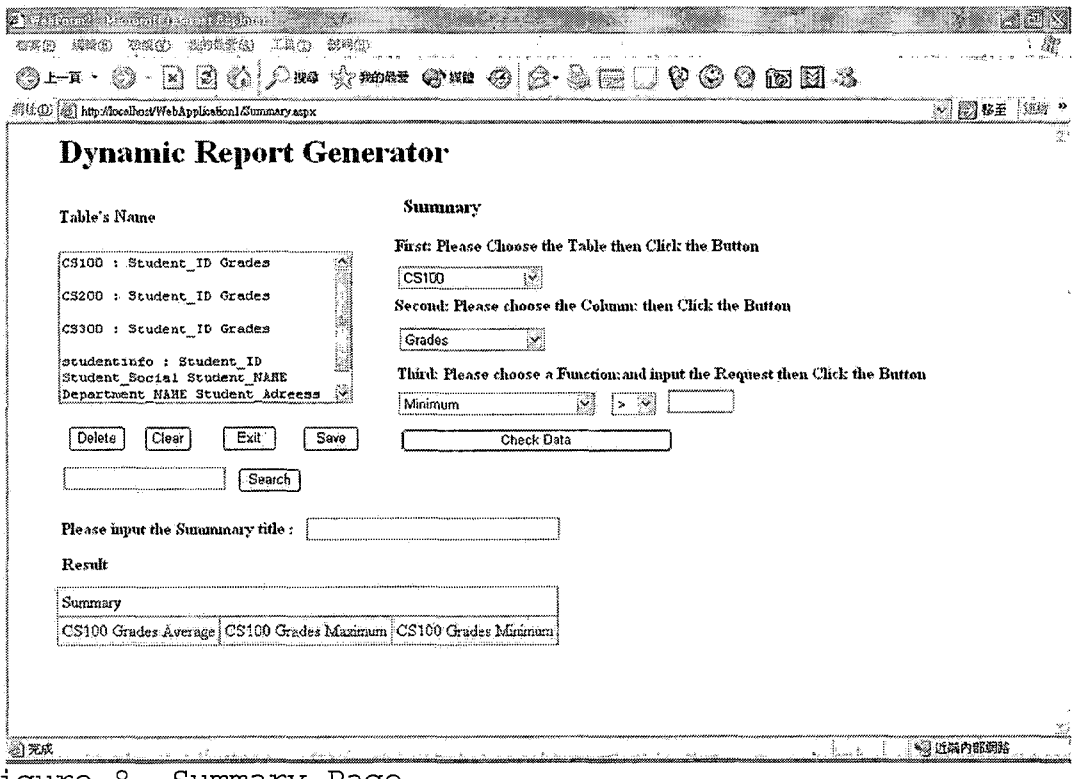

Figure 8. Summary Page

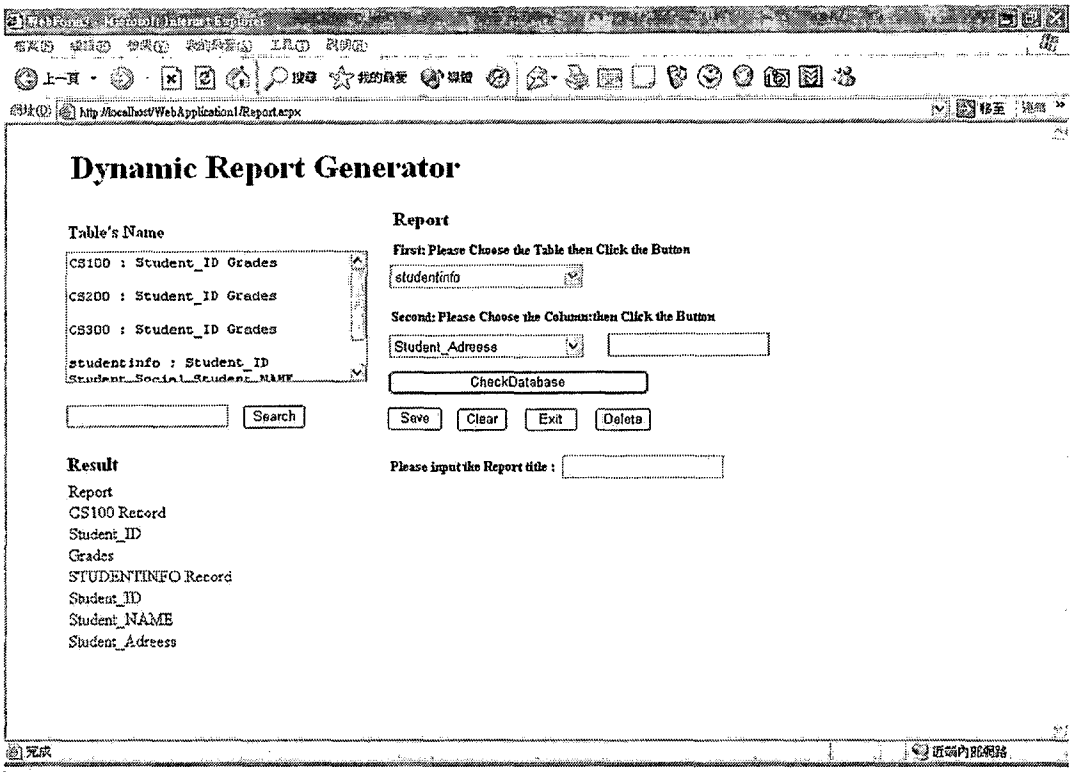

Figure 9. Report Page

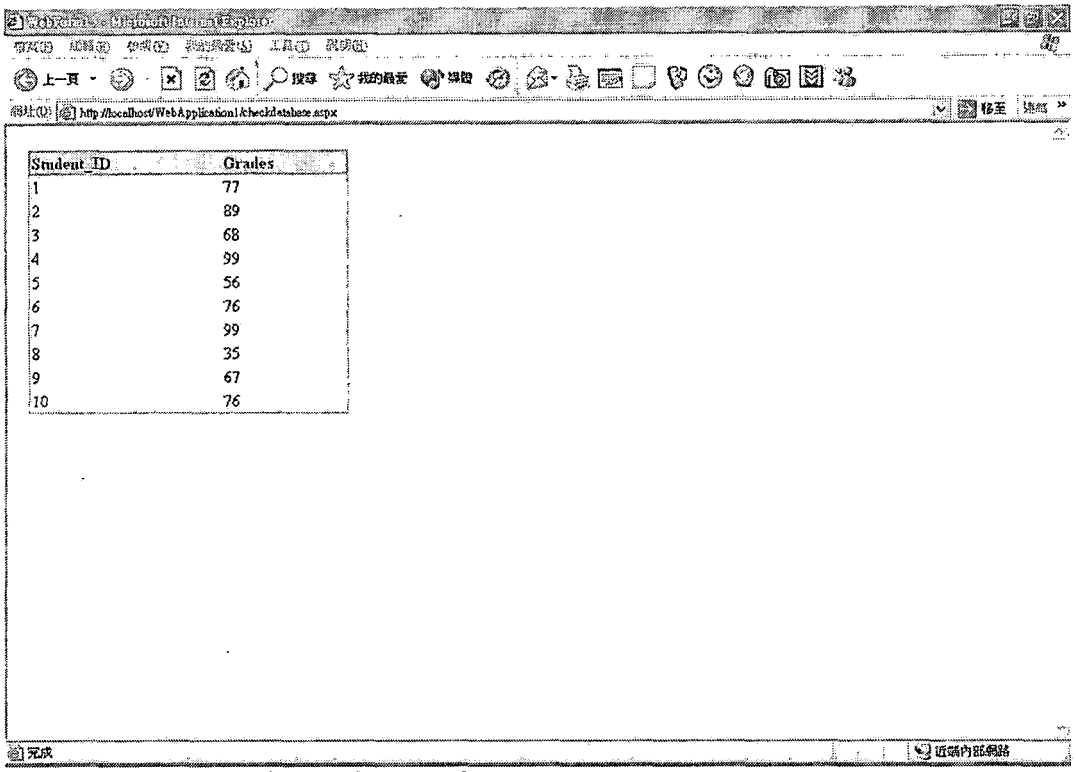

Figure 10. Result of Check Data Button

3.3.1.6 Continue Page. The DRG program has four options in this page.

a. The first option is go back to the database choose page. This option can let the user goes back to the database choose page directly if she want to generate another database.

b. The second option is go back to the Summary page. This option can let the user goes back to the Summary page directly if the user wants to do the Summary report again or change the content of the previous Summary report.

c. The third option is go back to the Report page. This option can let the user goes back to the Report page directly if the user wants to do the Report report again or change the content of the previous Report report.

d. The last option is exit and display the final report.

The user can choose one of these options that depend on user's next action. Then the user can press the confirm button to leave this page (see Figure 11).

3.3.1.7 Close Page. The DRG program displays the final report in this page. The user can see the layout of the Summary and the Report is itself in this page. The Summary report is immoveable. However, the Report's report is dynamically. The user can choose one of the tables and sort the table by every column. At last, the user can press the "Close the Window" button to close the whole window. At the same time, the temporary directory will been delete, too (see Figure 12).

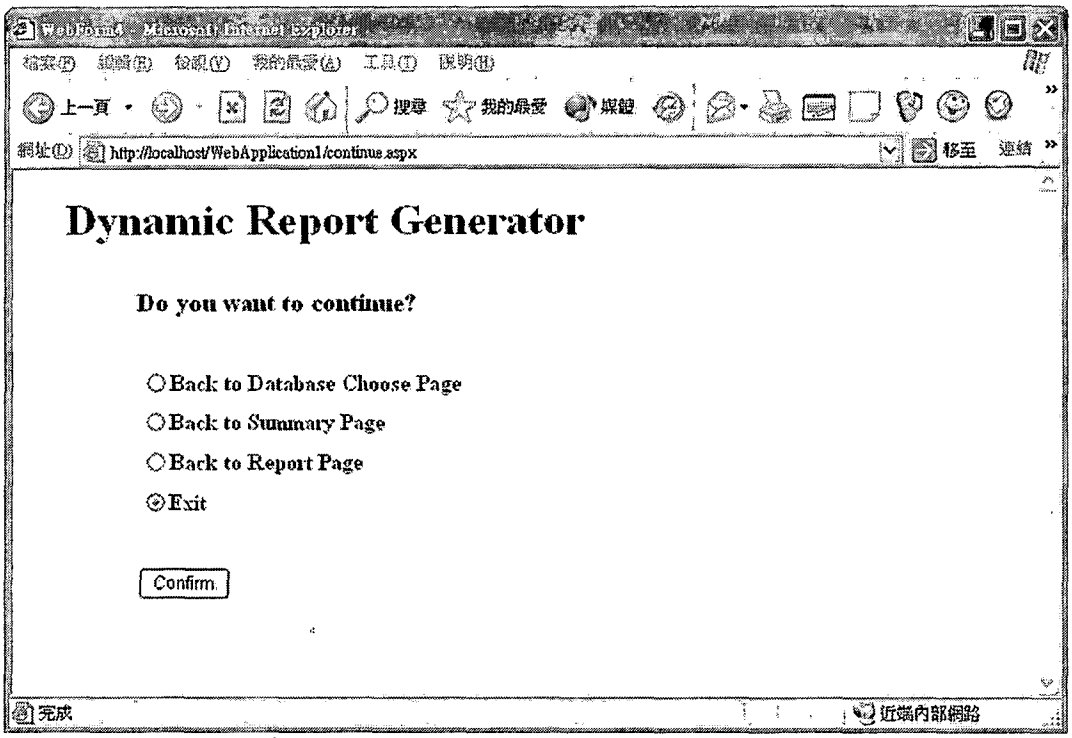

Figure 11. Continue Page

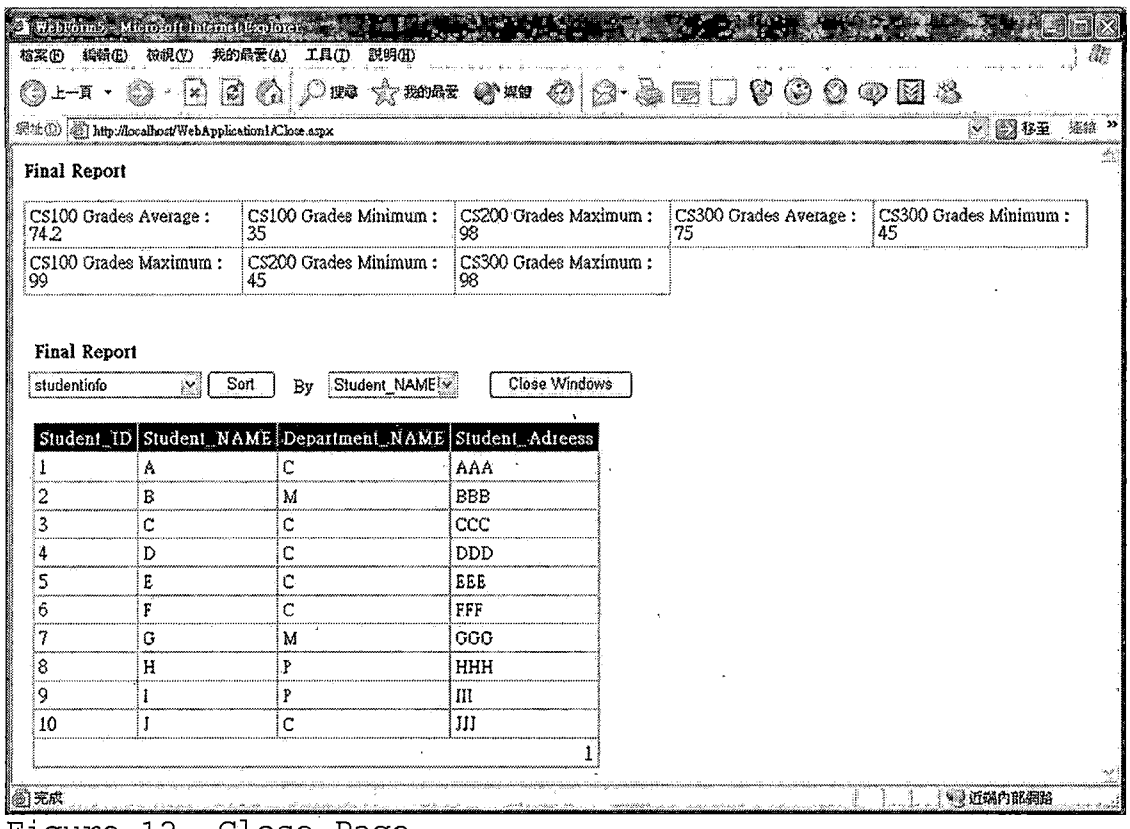

Figure 12. Close Page

CHAPTER FOUR

RESULTS

The DRG version written in ASP.NET was successful. It can receive the data from the Access databases, transform the data into an XML document, and produce an XML file. The DRG can access the data, generate the report from the XML document, and print outs the results in a new format. However, there is a problem working with the Access database. When there are more than 12 tables in an Access file, the program is unable to continue. The problem is that ASP.NET database has problems accepting table.

CHAPTER FIVE

SUMMARY

5.1 Summary

DRG is a dynamic Internet program. The user uses this program to get the data from the Internet and generates a report based on user's input. During the process, the user does not worry about the database type. The program's conversion module can access several different kinds of databases, and transform these databases into an XML document.

5.2 Conclusions

This project has proved that the user can use DRG to transform other database's data into an XML format, and generate a report based on user's command when the server's policy allows the client to access the server's database. User can print out the result report in a new layout. As XML becomes more popular, DRG should also become more popular because of its XML convert module.

However, DRG has two problems, which need to be solved or improved.

i. The limit on the number of tables in ASP.NET program.

ii. The server's policy.

There is one way to solve the first defect now. The user can separate the Access database's tables before she runs the program. When the user let table's number of the database is under the 12, DRG can avoid the error of the limit on the number of the table. The second defect is still unresolved unless a new server policy just for read access is designed. Before that, there is no directly way to solve this problem.

APPENDIX A

PROJECT CODE

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="choice1.aspx.vb"
Inherits="WebApplication1.WebForm9"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>WebForm9</title>
        <meta content="Microsoft Visual Studio .NET 7.0" name="GENERATOR">
        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
        <meta content="JavaScript" name="vs_defaultClientScript">
        <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    </HEAD>
    <body MS_POSITIONING="GridLayout">
        <form id="Form1" method="post" runat="server">
            <h1>Dynamic Report Generator</h1>
            <asp:textbox id="TextBox1" style="Z-INDEX: 101; LEFT: 31px; POSITION: absolute; TOP:
122px" runat="server" Width="325px"></asp:textbox><asp:label id="Label1" style="Z-INDEX: 102;
LEFT: 89px; POSITION: absolute; TOP: 99px" runat="server" Font-Bold="True">Please input the URL
Address</asp:label><asp:button id="Button1" style="Z-INDEX: 103; LEFT: 148px; POSITION: absolute;
TOP: 289px" runat="server" Text="Confirm"></asp:button><asp:label id="Label2" style="Z-INDEX: 104;
LEFT: 370px; POSITION: absolute; TOP: 127px" runat="server" Font-Bold="True"
ForeColor="Red">Please Input the URL Address</asp:label><asp:textbox id="username" style="Z-
INDEX: 105; LEFT: 33px; POSITION: absolute; TOP: 192px" runat="server"
Width="323px"></asp:textbox><asp:textbox id="password" style="Z-INDEX: 106; LEFT: 32px;
POSITION: absolute; TOP: 247px" runat="server" Width="326px"
TextMode="Password"></asp:textbox><asp:label id="Label3" style="Z-INDEX: 107; LEFT: 100px;
POSITION: absolute; TOP: 164px" runat="server" Width="211px" Font-Bold="True">Please input the
user name</asp:label><asp:label id="Label4" style="Z-INDEX: 108; LEFT: 107px; POSITION: absolute;
TOP: 225px" runat="server" Width="184px" Font-Bold="True">Please input the
password</asp:label></form>
    </body>
</HTML>
```

choice1.aspx.vb

```
Imports nsoftware
Imports System
Imports System.IO


Public Class WebForm9
    Inherits System.Web.UI.Page
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Requiredfieldvalidator1 As
System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Ftp1 As nsoftware.IPWorks.Ftp
    Protected WithEvents username As System.Web.UI.WebControls.TextBox
    Protected WithEvents password As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button

#Region " Web Form 設計工具產生的程式碼 "

    Private components As System.ComponentModel.IContainer
    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Me.Ftp1 = New nsoftware.IPWorks.Ftp(Me.components)
```

```
        'Ftp1
        '
        Me.Ftp1.About = ""
        Me.Ftp1.FirewallPort = 1080
        Me.Ftp1.FirewallType = nsoftware.IPWorks.FtpFirewallTypes.fwNone
        Me.Ftp1.TransferMode = nsoftware.IPWorks.FtpTransferModes.tmDefault

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Dim tempfilesname As New ArrayList()
    Dim tempfilessize As New ArrayList()
    Dim tempfilestime As New ArrayList()
    Private currentDir As System.IO.Directory

    Public Sub page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Label2.Text = ""
        Dim f As New System.IO.FileInfo(("c:\dygtemp\"))
        If f.Exists = True Then
            Dim dir As New DirectoryInfo(("c:\dygtemp\"))
            For Each f In dir.GetFiles("*.*")
                f.Delete()
            Next
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If TextBox1.Text = "" Then
            Label2.Text = "Please input the URL Address"
        ElseIf TextBox1.Text.Length < 7 Then
            Label2.Text = "Please input a standard URL address"
        Else
            Dim i
            Ftp1.TransferMode = nsoftware.IPWorks.FtpTransferModes.tmDefault
            BindData()
            currentDir.CreateDirectory(("c:\dygtemp\").ToString())
            currentDir.SetCurrentDirectory(("c:\dygtemp\").ToString())
            For i = 0 To tempfilesname.Count - 1
                Ftp1.LocalFile = currentDir.GetCurrentDirectory & "\" & tempfilesname.Item(i).ToString
                Ftp1.RemoteFile = tempfilesname.Item(i).ToString
                Ftp1.Download()
            Next
            Session("tempfilesname") = tempfilesname
            Session("tempfilessize") = tempfilessize
            Session("tempfilestime") = tempfilestime
            Server.Transfer("Choice2.aspx")
        End If

    End Sub
```

```
    Private Sub BindData()

        Ftp1.RemoteHost = TextBox1.Text
        Ftp1.User = username.Text
        Ftp1.Password = password.Text
        Ftp1.Logon()
        Ftp1.ListDirectoryLong()

    End Sub

    Private Sub Ftp1_OnDirList(ByVal sender As Object, ByVal e As IPWorks.FtpDirListEventArgs)
Handles Ftp1.OnDirList

        If Not e.IsDir Then
            tempfilesname.Add(e.FileName)
            tempfilessize.Add(e.FileSize)
            tempfilestime.Add(e.FileTime)
        End If
    End Sub
End Class
```

Choice2.aspx

```
<%@ Register TagPrefix="uc1" TagName="WebUserControl3" Src="WebUserControl3.ascx" %>
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="Choice2.aspx.vb"
Inherits="WebApplication1.WebForm1" aspCompat="True"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>WebForm1</title>
        <meta content="Microsoft Visual Studio .NET 7.0" name="GENERATOR">
        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
        <meta content="JavaScript" name="vs_defaultClientScript">
        <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    </HEAD>
    <body bgColor="#ffffff" MS_POSITIONING="GridLayout">
        <h1 id="Form1">Dynamic Report Generator</h1>
        <blockquote>
            <form runat="server">
                <blockquote>
                    <p></p>
                    <blockquote>
                        <p><asp:label id="Label2" style="Z-INDEX: 101; LEFT: 14px; POSITION:
absolute; TOP: 68px" runat="server" Font-Size="Medium" Font-Bold="True" Width="448px">Please
Click the database then push the Confirm Button</asp:label><asp:label id="Label1" style="Z-INDEX:
102; LEFT: 20px; POSITION: absolute; TOP: 108px" runat="server" Font-Bold="True">Database
List</asp:label><asp:listbox id="ListBox1" style="Z-INDEX: 103; LEFT: 21px; POSITION: absolute; TOP:
140px" runat="server" Width="445px"></asp:listbox><asp:label id="Label4" style="Z-INDEX: 104; LEFT:
294px; POSITION: absolute; TOP: 229px" runat="server" ForeColor="Red">Label</asp:label><asp:label
id="Label3" style="Z-INDEX: 105; LEFT: 25px; POSITION: absolute; TOP: 240px" runat="server" Font-
Bold="True">File datails</asp:label><asp:datalist id="DataList2" style="Z-INDEX: 106; LEFT: 25px;
POSITION: absolute; TOP: 268px" runat="server" Width="730px" BorderColor="#CC9966"
BorderStyle="None" BackColor="White" CellPadding="4" GridLines="Both" BorderWidth="1px">
                                <SelectedItemStyle Font-Bold="True" ForeColor="#663399"
BackColor="#FFCC66"></SelectedItemStyle>
                                <ItemStyle ForeColor="#330099" BackColor="White"></ItemStyle>
                                <HeaderTemplate>
                                    File List
                                </HeaderTemplate>
```

```
                        <ItemStyle ForeColor="#003399" BackColor="White"></ItemStyle>
                        <ItemTemplate>
                            <%# container.dataitem%>
                        </ItemTemplate>
                        <FooterStyle ForeColor="#330099" BackColor="#FFFFCC"></FooterStyle>
                        <HeaderStyle Font-Bold="True" ForeColor="#FFFFCC"
BackColor="#990000"></HeaderStyle>
                    </asp:datalist><asp:button id="Button1" style="Z-INDEX: 107; LEFT: 195px;
POSITION: absolute; TOP: 227px" runat="server" Text="Confirm"></asp:button></p>
                </blockquote></blockquote>
            </form>
        </blockquote>
    </body>
</HTML>
```

Choice2.aspx.vb

```
Imports System.Data
Imports System.Data.OleDb
Imports System.IO
Imports System.Text
Imports System.Collections
Imports System.ComponentModel

Public Class WebForm1
    Inherits System.Web.UI.Page
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Requiredfieldvalidator1 As
System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents ListBox1 As System.Web.UI.WebControls.ListBox
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents DataList1 As System.Web.UI.WebControls.DataList
    Protected WithEvents DataList2 As System.Web.UI.WebControls.DataList
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Requiredfieldvalidator2 As
System.Web.UI.WebControls.RequiredFieldValidator

#Region
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        InitializeComponent()
    End Sub
#End Region
    Dim dir As DirectoryInfo
    Dim strdir As String
    Dim dataarray As New ArrayList()

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        addname1()
        addname2()
        addname3()
        Dim inputurl
        Dim dir As DirectoryInfo
        Dim strdir As String
```

```
        Dim pathname As Array
        Dim i, j
        inputurl = Session("inputurl")
        If Not Page.IsPostBack Then
            pathname = inputurl.ToString.Split("/")Imports System.Data
Imports System.Data.OleDb
Imports System.IO
Imports System.Text
Imports System.Collections
Imports System.ComponentModel

Public Class WebForm1
    Inherits System.Web.UI.Page
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Requiredfieldvalidator1 As
System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents ListBox1 As System.Web.UI.WebControls.ListBox
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents DataList1 As System.Web.UI.WebControls.DataList
    Protected WithEvents DataList2 As System.Web.UI.WebControls.DataList
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Ftp1 As nsoftware.IPWorks.Ftp
    Private components As System.ComponentModel.IContainer
    Protected WithEvents Requiredfieldvalidator2 As
System.Web.UI.WebControls.RequiredFieldValidator

#Region " Web Form 設計工具產生的程式碼 "

    '此呼叫為 Web Form 設計工具的必要項。
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Me.Ftp1 = New nsoftware.IPWorks.Ftp(Me.components)
        '
        'Ftp1
        '
        Me.Ftp1.About = ""
        Me.Ftp1.FirewallPort = 1080
        Me.Ftp1.FirewallType = nsoftware.IPWorks.FtpFirewallTypes.fwNone
        Me.Ftp1.TransferMode = nsoftware.IPWorks.FtpTransferModes.tmDefault

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        'CODEGEN: 此方法呼叫為 Web Form 設計工具的必要項
        '請勿使用程式碼編輯器來修改它。
        InitializeComponent()
    End Sub

#End Region
    Dim dir As DirectoryInfo
    Dim strdir As String
    Dim dataarray As New ArrayList()
```

```
  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    addname1()
    addname2()
    addname3()
    Dim strdir As String
    Dim pathname As Array
    Dim i
    Label4.Text = ""

    If Not Page.IsPostBack Then
      For i = 0 To dataarray.Count - 1
        getfilename(dataarray(i))
      Next
    End If
  End Sub

  Dim datachoose, datachoose2 As String
  Dim tablelist As ArrayList
  Dim mydataset As New DataSet()

  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

    If ListBox1.SelectedItem Is Nothing Then
      Label4.Text = "(Please choose the database)"
    Else
      datachoose = ListBox1.SelectedItem.Text
      datachoose2 = ListBox1.SelectedItem.Text
      Dim title = datachoose.ToCharArray(datachoose.LastIndexOf(".") + 1, 3)

      Select Case title
        Case Is = "mdb"
          getaccessdata()
        Case Is = "xml"
          getxml()
        Case Is = "sql"
          getoracle()
      End Select
      Response.Redirect("choose3.aspx")
    End If

  End Sub

  Public Function getfilename(ByVal s As String)
    Dim datalist As New ArrayList()
    If Not Session("datalist") Is Nothing Then
      datalist = Session("datalist")
    End If
    Dim i
    Dim tempfilesname As New ArrayList()
    Dim tempfilessize As New ArrayList()
    Dim tempfilestime As New ArrayList()
    tempfilesname = Session("tempfilesname")
    tempfilessize = Session("tempfilessize")
    tempfilestime = Session("tempfilestime")
    For i = 0 To tempfilesname.Count - 1
      If tempfilesname.Item(i).ToString.EndsWith(s) Then
        ListBox1.Items.Add(tempfilesname.Item(i))
```

```
            datalist.Add("Name: " & tempfilesname.Item(i) & vbTab & "   Size: " & vbTab &
tempfilessize.Item(i) & vbTab & "   Time: " & vbTab & tempfilestime.Item(i))
          End If
        Next
        Session("datalist") = datalist
        DataList2.DataSource = datalist
        DataList2.DataBind()
    End Function

    Public Function addname1()
        dataarray.Add("mdb")
    End Function

    Public Function addname2()
        dataarray.Add("xml")
    End Function

    Public Function addname3()
        dataarray.Add("sql")
    End Function

    Public Function getaccessdata()
        Dim title = datachoose.ToCharArray(datachoose.LastIndexOf(".") + 1, 3)
        ListBox1.Items.Add(title)
        Dim objCat
        Dim objTables
        Dim objTable

        objCat = Server.CreateObject("ADOX.Catalog")
        objCat.ActiveConnection = "provider=microsoft.jet.oledb.4.0;" & "data source=" & ("c:\dygtemp\" +
datachoose)

        objTables = objCat.Tables

        For Each objTable In objTables

            If objTable.Type = "TABLE" Then
                Dim connstr As String = "provider=microsoft.jet.oledb.4.0;data source=" & ("c:\dygtemp\" +
datachoose)
                Dim conn As OleDbConnection, adap As OleDbDataAdapter
                conn = New OleDbConnection(connstr)
                conn.Open()
                Dim sql = "select * from " & objTable.Name
                adap = New OleDbDataAdapter(sql, conn)
                adap.Fill(mydataset, objTable.Name)
                mydataset.WriteXml(("c:\dygtemp\" + "temp.xml"), XmlWriteMode.WriteSchema)
                conn.Close()
            End If
        Next
        objCat = Nothing
        datachoose = "temp.xml"
        Session("datachoose") = datachoose
        Session("datachoose2") = datachoose2
    End Function

    Public Function getxml()
        datachoose = ListBox1.SelectedItem.Text
        Session("datachoose") = datachoose
    End Function
```

```
  Public Function getoracle()
    datachoose = ListBox1.SelectedItem.Text
    datachoose2 = ListBox1.SelectedItem.Text
    Dim objCat
    Dim objTables
    Dim objTable
    objCat = Server.CreateObject("ADOX.Catalog")
    objCat.ActiveConnection = "provider=MSDAORA;" & "data source=" & ("c:\dygtemp\" +
datachoose)

    objTables = objCat.Tables

    For Each objTable In objTables
      If objTable.Type = "TABLE" Then
        Dim connstr As String = "provider=MSDAORA;data source=" & ("c:\dygtemp\" + datachoose)
        Dim conn As OleDbConnection, adap As OleDbDataAdapter
        conn = New OleDbConnection(connstr)
        conn.Open()
        Dim sql = "select * from " & objTable.Name
        adap = New OleDbDataAdapter(sql, conn)
        adap.Fill(mydataset, objTable.Name)
        mydataset.WriteXml(("c:\dygtemp\" + "temp.xml"), XmlWriteMode.WriteSchema)
        conn.Close()
      End If
    Next

    objCat = Nothing

    datachoose = "temp.xml"
    Session("datachoose") = datachoose
    Session("datachoose2") = datachoose2
  End Function

End Class
```

Choose3.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="choose3.aspx.vb"
Inherits="WebApplication1.WebForm7"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>WebForm7</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.0">
    <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
  </HEAD>
  <body MS_POSITIONING="GridLayout">
    <form id="Form7" method="post" runat="server">
      <h1>Dynamic Report Generator</h1>
      <Asp:Button id="Button1" style="Z-INDEX: 105; LEFT: 154px; POSITION: absolute; TOP:
282px" runat="server" Text="Confirm"></Asp:Button>
      <FONT face="新細明體">
        <asp:RadioButtonList id="choice" style="Z-INDEX: 104; LEFT: 146px; POSITION:
absolute; TOP: 164px" runat="server" Height="97px">
          <asp:ListItem Value="s" Selected="True">Summary</asp:ListItem>
          <asp:ListItem Value="r">Report</asp:ListItem>
        </asp:RadioButtonList>
```

```
            <asp:Label id="Label1" style="Z-INDEX: 103; LEFT: 72px; POSITION: absolute; TOP:
114px" runat="server" Font-Bold="True" Font-Size="Medium" Width="222px">Please Choose the
operation</asp:Label></FONT></form>
    </body>
</HTML>
```

choose3.aspx.vb

```
Public Class WebForm7
   Inherits System.Web.UI.Page
   Protected WithEvents Label1 As System.Web.UI.WebControls.Label
   Protected WithEvents Button1 As System.Web.UI.WebControls.Button
   Protected WithEvents choice As System.Web.UI.WebControls.RadioButtonList

#Region
   <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
   End Sub
   Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
      InitializeComponent()
   End Sub
#End Region

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
      If choice.SelectedItem.Text = "Summary" Then
         Server.Transfer("Summary.aspx")
      Else
         Server.Transfer("Report.aspx")
      End If
   End Sub
End Class
```

Summary.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="Summary.aspx.vb"
Inherits="WebApplication1.WebForm2" aspCompat="True"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>WebForm2</title>
        <meta content="Microsoft Visual Studio .NET 7.0" name="GENERATOR">
        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
        <meta content="JavaScript" name="vs_defaultClientScript">
        <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    </HEAD>
    <body MS_POSITIONING="GridLayout">
        <form id="Form1" method="post" runat="server">
            <h1><asp:label id="Label1" style="Z-INDEX: 101; LEFT: 49px; POSITION: absolute; TOP:
9px" runat="server">Dynamic Report Generator</asp:label></h1>
            <asp:button id="saveButton" style="Z-INDEX: 102; LEFT: 283px; POSITION: absolute; TOP:
296px" runat="server" Text=" Save "></asp:button><asp:button id="exitButton" style="Z-INDEX: 103;
LEFT: 206px; POSITION: absolute; TOP: 296px" runat="server" Text=" Exit "
Width="52px"></asp:button>
            <h3><asp:label id="Label2" style="Z-INDEX: 104; LEFT: 380px; POSITION: absolute; TOP:
72px" runat="server">Summary</asp:label></h3>
            <asp:label id="Label3" style="Z-INDEX: 105; LEFT: 374px; POSITION: absolute; TOP:
235px" runat="server" Width="510px" Font-Bold="True">Third: Please choose a Function:and input the
Request then Click the Button</asp:label><asp:label id="Label4" style="Z-INDEX: 106; LEFT: 372px;
POSITION: absolute; TOP: 168px" runat="server" Width="389px" Font-Bold="True">Second: Please
```

```
choose the Column: then Click the Button</asp:label></B5><asp:dropdownlist id="funct" style="Z-
INDEX: 107; LEFT: 376px; POSITION: absolute; TOP: 263px" runat="server" Width="189px">
                <asp:ListItem Value="Average" Selected="True">Average</asp:ListItem>
                <asp:ListItem Value="Maximum">Maximum</asp:ListItem>
                <asp:ListItem Value="Minimum">Minimum</asp:ListItem>
                <asp:ListItem Value="Count">Count</asp:ListItem>
            </asp:dropdownlist>
            <h4><asp:label id="Label6" style="Z-INDEX: 108; LEFT: 49px; POSITION: absolute; TOP:
82px" runat="server">Table's Name</asp:label>
                <asp:textbox id="Searchbox2" style="Z-INDEX: 119; LEFT: 54px; POSITION: absolute;
TOP: 336px" runat="server"></asp:textbox>
                <asp:button id="SummarySearch" style="Z-INDEX: 118; LEFT: 221px; POSITION:
absolute; TOP: 337px" runat="server" Text="Search"></asp:button>
                <asp:label id="Label20" style="Z-INDEX: 116; LEFT: 372px; POSITION: absolute; TOP:
110px" runat="server" Font-Bold="True">First: Please Choose the Table then Click the
Button</asp:label><asp:label id="Label7" style="Z-INDEX: 109; LEFT: 52px; POSITION: absolute; TOP:
422px" runat="server">Result</asp:label><asp:textbox id="TextBox1" style="Z-INDEX: 110; LEFT: 49px;
POSITION: absolute; TOP: 125px" runat="server" Width="282px" ReadOnly="True"
TextMode="MultiLine" Height="148px" Rows="10"></asp:textbox>
                <asp:DropDownList id="DropDownList5" style="Z-INDEX: 111; LEFT: 376px; POSITION:
absolute; TOP: 140px" runat="server" AutoPostBack="True" Width="139px"
OnSelectedIndexChanged="Index_Changed"></asp:DropDownList>
                <asp:DropDownList id="DropDownList6" style="Z-INDEX: 112; LEFT: 377px; POSITION:
absolute; TOP: 200px" runat="server" AutoPostBack="True" Width="140px"
OnSelectedIndexChanged="Index_Changed1"></asp:DropDownList>
                <asp:DataList id="DataList1" style="Z-INDEX: 113; LEFT: 47px; POSITION: absolute;
TOP: 453px" runat="server" BorderColor="#CC9966" BorderStyle="None" BackColor="White"
CellPadding="4" GridLines="Both" BorderWidth="1px" RepeatColumns="3">
                    <HeaderTemplate>
                        Summary
                    </HeaderTemplate>
                    <AlternatingItemStyle BackColor="#F7F7F7"></AlternatingItemStyle>
                    <ItemStyle ForeColor="#4A3C8C" BackColor="#E7E7FF"></ItemStyle>
                    <ItemTemplate>
                        <%# container.dataitem%>
                    </ItemTemplate>
                </asp:DataList>
                <asp:Button id="Clear2" style="Z-INDEX: 114; LEFT: 132px; POSITION: absolute; TOP:
296px" runat="server" Text="Clear"></asp:Button>
                <asp:Button id="Delete2" style="Z-INDEX: 115; LEFT: 59px; POSITION: absolute; TOP:
296px" runat="server" Text="Delete"></asp:Button>
                <asp:Button id="Checkdata" style="Z-INDEX: 117; LEFT: 377px; POSITION: absolute;
TOP: 298px" runat="server" Text="Check Data" Width="265px"></asp:Button>
                <asp:Label id="Label13" style="Z-INDEX: 120; LEFT: 551px; POSITION: absolute; TOP:
140px" runat="server" ForeColor="Red">Label</asp:Label>
                <asp:Label id="Label14" style="Z-INDEX: 121; LEFT: 552px; POSITION: absolute; TOP:
202px" runat="server" ForeColor="Red">Label</asp:Label>
                <asp:Label id="Label15" style="Z-INDEX: 122; LEFT: 377px; POSITION: absolute; TOP:
343px" runat="server" ForeColor="Red">Label</asp:Label>
                <asp:DropDownList id="DropDownList4" style="Z-INDEX: 123; LEFT: 582px; POSITION:
absolute; TOP: 263px" runat="server" Width="42px">
                    <asp:ListItem Value="&gt;" Selected="True">&gt;</asp:ListItem>
                    <asp:ListItem Value="=">=</asp:ListItem>
                    <asp:ListItem Value="&lt;">&lt;</asp:ListItem>
                </asp:DropDownList>
                <asp:TextBox id="TextBox3" style="Z-INDEX: 124; LEFT: 636px; POSITION: absolute;
TOP: 261px" runat="server" Width="63px"></asp:TextBox>
                <asp:Label id="Label10" style="Z-INDEX: 125; LEFT: 51px; POSITION: absolute; TOP:
387px" runat="server">Please input the Summmary title : </asp:Label>
```

```
            <asp:TextBox id="summarytitletext" style="Z-INDEX: 126; LEFT: 287px; POSITION:
absolute; TOP: 386px" runat="server" Width="244px"></asp:TextBox></h4>
        </form>
    </body>
</HTML>
```

Summary.aspx.vb

```
Imports System
Imports System.IO
Imports System.Data
Imports System.Data.OleDb
Imports System.Xml
Imports System.Text
Imports System.Xml.XPath

Public Class WebForm2
    Inherits System.Web.UI.Page
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents funct As System.Web.UI.WebControls.DropDownList
    Protected WithEvents saveButton As System.Web.UI.WebControls.Button
    Protected WithEvents clearButton As System.Web.UI.WebControls.Button
    Protected WithEvents exitButton As System.Web.UI.WebControls.Button
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents searchBox As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents DropDownList5 As System.Web.UI.WebControls.DropDownList
    Protected WithEvents DropDownList6 As System.Web.UI.WebControls.DropDownList
    Protected WithEvents Label20 As System.Web.UI.WebControls.Label
    Protected WithEvents Searchbox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label13 As System.Web.UI.WebControls.Label
    Protected WithEvents Label14 As System.Web.UI.WebControls.Label
    Protected WithEvents Label15 As System.Web.UI.WebControls.Label
    Protected WithEvents Clear2 As System.Web.UI.WebControls.Button
    Protected WithEvents Delete2 As System.Web.UI.WebControls.Button
    Protected WithEvents Checkdata As System.Web.UI.WebControls.Button
    Protected WithEvents SummarySearch As System.Web.UI.WebControls.Button
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents TextBox3 As System.Web.UI.WebControls.TextBox
    Protected WithEvents DataList1 As System.Web.UI.WebControls.DataList
    Protected WithEvents DropDownList4 As System.Web.UI.WebControls.DropDownList
    Protected WithEvents Label10 As System.Web.UI.WebControls.Label
    Protected WithEvents summarytitletext As System.Web.UI.WebControls.TextBox
    Protected WithEvents labelva As System.Web.UI.WebControls.TextBox

#Region
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        InitializeComponent()
    End Sub
#End Region
```

```
    Dim i = 0
    Dim j = 0
    Dim k = 0
    Dim choosefunction
    Dim inputcolumn
    Dim choosefunctionresult
    Dim resultarray As New ArrayList()
    Dim resultarray1 As New ArrayList()
    Dim mutli1 = False
    Dim mutli2 = False
    Dim summarytitle

    Public Sub page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Label13.Text = ""
        Label14.Text = ""
        Label15.Text = ""
        Dim datachoose
        datachoose = Session("datachoose")
        If Not Page.IsPostBack Then
            TextBox1.Text = "" 'erase textbox1
            Dim xmldoc As New XmlDataDocument()
            xmldoc.DataSet.ReadXml(Server.MapPath(datachoose))
            Dim tablelist As New ArrayList()
            For i = 0 To xmldoc.DataSet.Tables.Count - 1
                TextBox1.Text &= xmldoc.DataSet.Tables(i).TableName.ToString & " : "
                For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                    TextBox1.Text &= xmldoc.DataSet.Tables(i).columns(j).columnname & " "
                Next
                TextBox1.Text &= vbNewLine
                TextBox1.Text &= vbNewLine
                tablelist.Add(xmldoc.DataSet.Tables(i).TableName)
            Next i
            DropDownList5.DataSource = tablelist
            DropDownList5.DataBind()
            Dim tablechoose As String
            tablechoose = DropDownList5.SelectedItem.Text
            Dim columnlist As New ArrayList()
            For i = 0 To xmldoc.DataSet.Tables.Count - 1
                If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
                    For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                        columnlist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
                    Next
                End If
            Next i
            DropDownList6.DataSource = columnlist
            DropDownList6.DataBind()
            Session("tablechoose") = tablechoose
            Dim columnchoose
            columnchoose = DropDownList6.SelectedItem.Text
            Session("columnchoose") = columnchoose
        End If
    End Sub

    Sub Index_Changed(ByVal sender As Object, ByVal e As EventArgs)
        Dim datachoose
        datachoose = Session("datachoose")
        Dim xmldoc As New XmlDataDocument()
        xmldoc.DataSet.ReadXml(Server.MapPath(datachoose))
        Dim tablechoose As String
```

```
        tablechoose = DropDownList5.SelectedItem.Text
        Dim columnlist As New ArrayList()
        For i = 0 To xmldoc.DataSet.Tables.Count - 1
            If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
                For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                    columnlist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
                Next
            End If
        Next i
        DropDownList6.DataSource = columnlist
        DropDownList6.DataBind()
        Session("tablechoose") = tablechoose
        Dim columnchoose
        columnchoose = DropDownList6.SelectedItem.Text
        Session("columnchoose") = columnchoose
    End Sub

    Sub Index_Changed1(ByVal sender As Object, ByVal e As EventArgs)
        Dim columnchoose
        columnchoose = DropDownList6.SelectedItem.Text
        Session("columnchoose") = columnchoose
    End Sub

    Public Sub saveButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles saveButton.Click
        Dim tablechoose
        Dim columnchoose
        Dim datachoose
        datachoose = Session("datachoose")
        tablechoose = Session("tablechoose")
        columnchoose = Session("columnchoose")
        If Session("oldtablechoose") = Session("tablechoose") And Session("oldcolumnchoose") =
Session("columnchoose") And Session("oldchoosefunction") = funct.SelectedItem.Text Then
            Label15.Text = "You Had Repeat the Same Column or the same Function"
        Else
            'make dataset
            Dim mydataset As New DataSet()
            'input the xml document into the dataset
            mydataset.ReadXml(Server.MapPath(datachoose), XmlReadMode.Auto)

            Dim mydatatable As New DataTable()
            mydatatable = mydataset.Tables(tablechoose.ToString)
            Dim choosefunction
            choosefunction = funct.SelectedItem.Text
            If mydatatable.Rows.Count = 0 Then
                Label14.Text = "This Column is Empty"
            ElseIf Not mydatatable.Columns(columnchoose).datatype Is GetType(System.Int32) Then
                Label14.Text = "This Column's data can not be calculate."
            Else
                'Average function
                If choosefunction = "Average" Then
                    Dim sum = 0
                    For i = 0 To mydatatable.Rows.Count - 1
                        sum = sum + mydatatable.Rows(i).Item(columnchoose.ToString)
                    Next
                    sum = sum / (mydatatable.Rows.Count)
                    choosefunction = "Average"
                    choosefunctionresult = sum
                    'Maximum function
                ElseIf choosefunction = "Maximum" Then
```

```
            Dim max = 0
            For i = 0 To mydatatable.Rows.Count - 1
              If max < mydatatable.Rows(i).Item(columnchoose.ToString) Then max =
mydatatable.Rows(i).Item(columnchoose.ToString)
            Next
            choosefunction = "Maximum"
            choosefunctionresult = max
            'Minimum function
          ElseIf choosefunction = "Minimum" Then
            Dim min = 999999999
            For i = 0 To mydatatable.Rows.Count - 1
              If min > mydatatable.Rows(i).Item(columnchoose.ToString) Then min =
mydatatable.Rows(i).Item(columnchoose.ToString)
            Next
            choosefunction = "Minimum"
            choosefunctionresult = min
            'Count function
          ElseIf choosefunction = "Count" Then
            Dim mycount = 0
            If TextBox3.Text = "" Then
              Label15.Text = "Please Innput the Request in the Box"
            Else
              mydatatable.DefaultView.RowFilter = columnchoose.ToString &
DropDownList4.SelectedItem.Text & TextBox3.Text
              mycount = mydatatable.DefaultView.Count
              choosefunction = "Count"
              choosefunctionresult = mycount
            End If
          End If
          'check session, to see is it empty
          If Not Session("summaryresult") Is Nothing Then
            resultarray = Session("summaryresult")
          End If
          If Not Session("summaryresult1") Is Nothing Then
            resultarray1 = Session("summaryresult1")
          End If
          'print out result
          If choosefunction = "Count" And TextBox3.Text = "" Then
            Label15.Text = "Please Innput the Request in the Box"
          Else
            resultarray.Add(tablechoose & " " & columnchoose & " " & choosefunction)
            resultarray1.Add(tablechoose & " " & columnchoose & " " & choosefunction & " : " &
choosefunctionresult)
            DataList1.DataSource = resultarray
            DataList1.DataBind()
            Session("summaryresult") = resultarray
            Session("summaryresult1") = resultarray1
          End If
          Session("oldtablechoose") = tablechoose
          Session("oldcolumnchoose") = columnchoose
          Session("oldchoosefunction") = choosefunction
        End If
      End If
    End Sub

    'exit this page
    Private Sub exitButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles exitButton.Click
      If summarytitletext.Text = "" Then
        summarytitle = "Final Report"
```

```
        Else
            summarytitle = summarytitletext.Text
        End If
        Session("summarytitle") = summarytitle
        Response.Redirect("continue.aspx")
    End Sub
    'Clear the datalist
    Private Sub Clear2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Clear2.Click
        resultarray.Clear()
        resultarray.Add("It is Empty NOW!!!")
        DataList1.DataSource = resultarray
        DataList1.DataBind()
        resultarray.Clear()
        Session("summaryresult") = resultarray
    End Sub
    'delete from the datalist
    Private Sub Delete2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Delete2.Click
        If Not Session("summaryresult") Is Nothing Then
            resultarray = Session("summaryresult")
        End If
        If resultarray.Count = 0 Then
            Label15.Text = "No Item need to be Delete"
            resultarray.Add("It is Empty NOW!!!")
            DataList1.DataSource = resultarray
            DataList1.DataBind()
            resultarray.Clear()
            Session("summaryresult") = resultarray
        Else
            j = TextBox3.Text
            If j = Nothing Then
                j = resultarray.Count
                resultarray.RemoveAt(j - 1)
                DataList1.DataSource = resultarray
                DataList1.DataBind()
                Session("summaryresult") = resultarray
            ElseIf j > resultarray.Count Then
                Label15.Text = "The number in option box is too big"
            Else
                resultarray.RemoveAt(j - 1)
                DataList1.DataSource = resultarray
                DataList1.DataBind()
                Session("summaryresult") = resultarray
            End If
        End If
    End Sub

    Private Sub SummarySearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles SummarySearch.Click
        Dim datachoose2, columnchoose, tablechoose
        Dim searchdataset As New DataSet()
        datachoose2 = Session("datachoose2")
        columnchoose = Session("columnchoose")
        tablechoose = Session("tablechoose")
        Dim objCat
        Dim objTables
        Dim objTable
        Dim objColumn
        objCat = Server.CreateObject("ADOX.Catalog")
```

```
        objCat.ActiveConnection = "provider=microsoft.jet.oledb.4.0;" & "data source=" &
Server.MapPath(datachoose2)
        objTables = objCat.Tables
        ' For Each objTable In objTables
        Dim connstr As String = "provider=microsoft.jet.oledb.4.0;data source=" &
Server.MapPath(datachoose2)
        Dim conn As OleDbConnection, adap As OleDbDataAdapter
        conn = New OleDbConnection(connstr)
        conn.Open()
        Dim sql
        'For Each objColumn In objTable.columns
        If Searchbox2.Text.ToString.Substring(0, 1) = "*" Then
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '%" &
Searchbox2.Text.ToString.Substring(1, Searchbox2.Text.ToString.Length - 1) & "'"
        ElseIf Searchbox2.Text.ToString.Substring(Searchbox2.Text.ToString.Length - 1, 1) = "*" Then
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '" &
Searchbox2.Text.ToString.Substring(0, Searchbox2.Text.ToString.Length - 1) & "%'"
        ElseIf Searchbox2.Text.ToString.Substring(0, 1) = "*" And
Searchbox2.Text.ToString.Substring(Searchbox2.Text.ToString.Length - 1, 1) = "*" Then
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '%" &
Searchbox2.Text.ToString.Substring(1, Searchbox2.Text.ToString.Length - 2) & "%'"
        Else
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE " & Searchbox2.Text
& "'"
        End If
        'Next
        adap = New OleDbDataAdapter(sql, conn)
        adap.Fill(searchdataset, tablechoose)
        conn.Close()
        ' Next
        If searchdataset.Tables(0).Rows.Count = 0 Then
            Label15.Text = "Sorry, There is no result for your search."
        Else
            choosefunction = " The Search Result for " & Searchbox2.Text
            choosefunctionresult = searchdataset.Tables(0).Rows.Count
            'check session, to see is it empty
            If Not Session("summaryresult") Is Nothing Then
                resultarray = Session("summaryresult")
            End If

            If Not Session("summaryresult1") Is Nothing Then
                resultarray1 = Session("summaryresult1")
            End If
            'print out result
            resultarray.Add(choosefunction)
            resultarray1.Add(choosefunction & " : " & choosefunctionresult)
            DataList1.DataSource = resultarray
            DataList1.DataBind()
            Session("summaryresult") = resultarray
            Session("summaryresult1") = resultarray1
        End If
    End Sub

    Private Sub Checkdata_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Checkdata.Click
        Response.Redirect("checkdatabase.aspx")
    End Sub
End Class
```

Report.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="Report.aspx.vb"
Inherits="WebApplication1.WebForm3" aspCompat="True" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>WebForm3</title>
        <meta content="Microsoft Visual Studio .NET 7.0" name="GENERATOR">
        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
        <meta content="JavaScript" name="vs_defaultClientScript">
        <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    </HEAD>
    <body MS_POSITIONING="GridLayout">
        <form id="Form1" method="post" runat="server">
            <h1><asp:label id="Label1" style="Z-INDEX: 107; LEFT: 61px; POSITION: absolute; TOP:
24px" runat="server">Dynamic Report Generator</asp:label></h1>
            <asp:button id="savebutton" style="Z-INDEX: 101; LEFT: 366px; POSITION: absolute; TOP:
274px" runat="server" Text=" Save "></asp:button><asp:button id="clear" style="Z-INDEX: 102; LEFT:
433px; POSITION: absolute; TOP: 275px" runat="server" Text="Clear"
Width="50px"></asp:button><asp:button id="exitbutton" style="Z-INDEX: 103; LEFT: 499px; POSITION:
absolute; TOP: 275px" runat="server" Text=" Exit " Width="52px"></asp:button>
            <H3><asp:label id="Label2" style="Z-INDEX: 104; LEFT: 372px; POSITION: absolute; TOP:
84px" runat="server">Report</asp:label></H3>
            <h5><asp:label id="Label6" style="Z-INDEX: 105; LEFT: 59px; POSITION: absolute; TOP:
96px" runat="server" Font-Size="Small">Table's Name</asp:label><asp:label id="Label4" style="Z-
INDEX: 114; LEFT: 370px; POSITION: absolute; TOP: 180px" runat="server" Font-Bold="True">Second:
Please Choose the Column:then Click the Button</asp:label><asp:label id="Label8" style="Z-INDEX:
111; LEFT: 372px; POSITION: absolute; TOP: 116px" runat="server">First: Please Choose the Table
then Click the Button</asp:label><asp:dropdownlist id="DropDownList2" style="Z-INDEX: 112; LEFT:
369px; POSITION: absolute; TOP: 138px" runat="server" Width="186px" AutoPostBack="true"
OnSelectedIndexChanged="Index_Changed"></asp:dropdownlist><asp:dropdownlist
id="DropDownList3" style="Z-INDEX: 113; LEFT: 368px; POSITION: absolute; TOP: 205px"
runat="server" Width="188px" AutoPostBack="true"
OnSelectedIndexChanged="Index_Changed1"></asp:dropdownlist><asp:label id="Label7" style="Z-
INDEX: 106; LEFT: 59px; POSITION: absolute; TOP: 320px" runat="server" Width="58px" Font-
Size="Medium" Height="23px">Result</asp:label><asp:textbox id="TextBox1" style="Z-INDEX: 108;
LEFT: 57px; POSITION: absolute; TOP: 124px" runat="server" Width="290px" Height="126px"
ReadOnly="True" TextMode="MultiLine"></asp:textbox><asp:datalist id="DataList1" style="Z-INDEX:
109; LEFT: 59px; POSITION: absolute; TOP: 346px" runat="server" Width="358px">
                <HeaderTemplate>
                    Report
                </HeaderTemplate>
                <AlternatingItemStyle BackColor="#F7F7F7"></AlternatingItemStyle>
                <ItemStyle ForeColor="#4A3C8C" BackColor="#E7E7FF"></ItemStyle>
                <ItemTemplate>
                    <%# container.dataitem%>
                </ItemTemplate>
            </asp:datalist><asp:button id="Delete" style="Z-INDEX: 110; LEFT: 567px; POSITION:
absolute; TOP: 275px" runat="server" Text="Delete"></asp:button><asp:label id="Label5" style="Z-
INDEX: 115; LEFT: 637px; POSITION: absolute; TOP: 279px" runat="server"
ForeColor="Red">Label</asp:label><asp:textbox id="TextBox3" style="Z-INDEX: 116; LEFT: 580px;
POSITION: absolute; TOP: 204px" runat="server"></asp:textbox><asp:textbox id="Searchbox2"
style="Z-INDEX: 117; LEFT: 58px; POSITION: absolute; TOP: 272px"
runat="server"></asp:textbox><asp:button id="ReportSearch" style="Z-INDEX: 118; LEFT: 227px;
POSITION: absolute; TOP: 272px" runat="server" Text="Search"></asp:button><asp:datagrid
id="DataGrid2" style="Z-INDEX: 119; LEFT: 483px; POSITION: absolute; TOP: 365px"
runat="server"></asp:datagrid><asp:button id="checkbutton" style="Z-INDEX: 120; LEFT: 366px;
POSITION: absolute; TOP: 239px" runat="server" Text="CheckDatabase" Width="255px"></asp:button>
```

```
            <asp:Label id="Label3" style="Z-INDEX: 121; LEFT: 370px; POSITION: absolute; TOP:
325px" runat="server">Please input the Report title : </asp:Label>
            <asp:TextBox id="reporttitletext" style="Z-INDEX: 122; LEFT: 536px; POSITION: absolute;
TOP: 322px" runat="server"></asp:TextBox></h5>
        </form>
    </body>
</HTML>
```

Report.aspx.vb

```
Imports System
Imports System.IO
Imports System.Data
Imports System.Data.OleDb
Imports System.Xml
Imports System.Xml.XPath

Public Class WebForm3
    Inherits System.Web.UI.Page
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents savebutton As System.Web.UI.WebControls.Button
    Protected WithEvents clear As System.Web.UI.WebControls.Button
    Protected WithEvents exitbutton As System.Web.UI.WebControls.Button
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents DataGrid1 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents TextBox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents DataList1 As System.Web.UI.WebControls.DataList
    Protected WithEvents DropDownList3 As System.Web.UI.WebControls.DropDownList
    Protected WithEvents DropDownList2 As System.Web.UI.WebControls.DropDownList
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents Delete As System.Web.UI.WebControls.Button
    Protected WithEvents TextBox3 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Searchbox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents ReportSearch As System.Web.UI.WebControls.Button
    Protected WithEvents DataGrid2 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents checkbutton As System.Web.UI.WebControls.Button
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents reporttitletext As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label

#Region
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        InitializeComponent()
    End Sub
#End Region
    Private Sub page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Label5.Text = ""
        Dim datachoose
        datachoose = Session("datachoose")
        If Not Page.IsPostBack Then
            TextBox1.Text = "" 'erase textbox1
            Dim xmldoc As New XmlDataDocument()
```

```
        xmldoc.DataSet.ReadXml(Server.MapPath(datachoose))
        Dim tablelist As New ArrayList()
        For i = 0 To xmldoc.DataSet.Tables.Count - 1
            TextBox1.Text &= xmldoc.DataSet.Tables(i).TableName.ToString & " : "
            For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                TextBox1.Text &= xmldoc.DataSet.Tables(i).columns(j).columnname & " "
            Next
            TextBox1.Text &= vbNewLine
            TextBox1.Text &= vbNewLine
            tablelist.Add(xmldoc.DataSet.Tables(i).TableName)
        Next i
        DropDownList2.DataSource = tablelist
        DropDownList2.DataBind()

        Dim tablechoose As String
        tablechoose = DropDownList2.SelectedItem.Text
        Dim columnlist As New ArrayList()
        For i = 0 To xmldoc.DataSet.Tables.Count - 1
            If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
                For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                    columnlist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
                Next
            End If
        Next i
        DropDownList3.DataSource = columnlist
        DropDownList3.DataBind()

        Session("tablechoose") = tablechoose
        Dim columnchoose
        columnchoose = DropDownList3.SelectedItem.Text
        Session("columnchoose") = columnchoose
    End If
End Sub

Sub Index_Changed(ByVal sender As Object, ByVal e As EventArgs)
    Dim datachoose
    datachoose = Session("datachoose")
    Dim xmldoc As New XmlDataDocument()
    xmldoc.DataSet.ReadXml(Server.MapPath(datachoose))
    Dim tablechoose As String
    tablechoose = DropDownList2.SelectedItem.Text
    Dim columnlist As New ArrayList()
    For i = 0 To xmldoc.DataSet.Tables.Count - 1
        If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
            For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                columnlist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
            Next
        End If
    Next i
    DropDownList3.DataSource = columnlist
    DropDownList3.DataBind()
    Session("tablechoose") = tablechoose
    Dim columnchoose
    columnchoose = DropDownList3.SelectedItem.Text
    Session("columnchoose") = columnchoose
End Sub

Sub Index_Changed1(ByVal sender As Object, ByVal e As EventArgs)
    Dim columnchoose
    columnchoose = DropDownList3.SelectedItem.Text
```

```
    Session("columnchoose") = columnchoose
  End Sub

  Dim inputcondition
  Dim result
  Dim resultarray As New ArrayList()
  Dim i
  Dim j = 0
  Dim k
  Dim splitarray
  Dim tempdataset As New DataSet()
  Dim oldtablename
  Dim oldcolumnname
  Dim reporttitle

  Private Sub savebutton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles savebutton.Click
    Dim tablechoose
    Dim columnchoose
    Dim datachoose
    Dim tablenamelist As New ArrayList()
    Dim columnnamelist As New ArrayList()
    datachoose = Session("datachoose")
    tablechoose = Session("tablechoose")
    columnchoose = Session("columnchoose")
    If Not Session("tablenamelist") Is Nothing Then
      tablenamelist = Session("tablenamelist")
    End If
    ' reload reportresult, tempdataset, oldtablename
    If Not Session("reportresult") Is Nothing Then
      resultarray = Session("reportresult")
    End If
    If Not Session("tempdataset") Is Nothing Then
      tempdataset = Session("tempdataset")
    End If
    If Not Session("oldtablename") Is Nothing Then
      oldtablename = Session("oldtablename")
    End If
    If Not Session("oldcolumnname") Is Nothing Then
      oldcolumnname = Session("oldcolumnname")
    End If

    'make dataset
    Dim mydataset As New DataSet()
    'input the xml document into the dataset
    mydataset.ReadXml(Server.MapPath(datachoose), XmlReadMode.Auto)
    Dim mydatatable As New DataTable()
    mydatatable = mydataset.Tables(tablechoose.ToString)
    Dim temptable As New DataTable(tablechoose)
    If mydatatable.Rows.Count = 0 Then
      Label5.Text = "This Column is Empty"
    Else
      'first time input data
      If Not oldtablename = tablechoose Then
        resultarray.Add(mydatatable.TableName().ToString.ToUpper & "  " & "Record" & vbNewLine)
        resultarray.Add(columnchoose)
        tempdataset.Tables.Add(temptable)
        temptable.Columns.Add(columnchoose)
        For i = 0 To mydatatable.Rows.Count - 1
          Dim temprow As DataRow = temptable.NewRow
```

```
            temprow(columnchoose) = mydatatable.Rows(i).Item(columnchoose.ToString).ToString
            temptable.Rows.Add(temprow)
          Next
          tablenamelist.Add(tablechoose)
        Else ' not first time input data
          If Not columnchoose = oldcolumnname Then
            resultarray.Add(columnchoose)
            temptable = tempdataset.Tables(oldtablename)
            temptable.Columns.Add(columnchoose)
            For i = 0 To mydatatable.Rows.Count - 1
              temptable.Rows(i).Item(columnchoose.ToString) =
mydatatable.Rows(i).Item(columnchoose.ToString).ToString
            Next
          Else
            Label5.Text = "You had already choose this column."
          End If
        End If
        Session("tablenamelist") = tablenamelist
        oldtablename = tablechoose
        oldcolumnname = columnchoose
        Session("temptable") = temptable
        Session("oldtablename") = oldtablename
        Session("oldcolumnname") = oldcolumnname
        Session("reportresult") = resultarray
        Session("tempdataset") = tempdataset
        tempdataset.WriteXml(Server.MapPath("tempreport.xml"), XmlWriteMode.WriteSchema)
        DataList1.DataSource = resultarray
        DataList1.DataBind()
        'input element nto sort datalist
      End If
    End Sub

    Private Sub clear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
clear.Click
      Session("oldtablename") = ""
      Session("oldcolumnname") = ""
      Dim tablechoose = Session("tablechoose")
      tempdataset = Session("tempdataset")
      If Not tempdataset Is Nothing Then
        Dim temptable As DataTable = tempdataset.Tables(tablechoose)
        tempdataset.Clear()
        tempdataset.Tables.Clear()
        resultarray.Clear()
        resultarray.Add("It is Empty NOW!!!")
        DataList1.DataSource = resultarray
        DataList1.DataBind()
        resultarray.Clear()
        Session("reportresult") = resultarray
        tempdataset.WriteXml(Server.MapPath("tempreport.xml"), XmlWriteMode.WriteSchema)
        Session("tempdataset") = tempdataset
      Else
        resultarray.Add("It is Empty NOW!!!")
        DataList1.DataSource = resultarray
        DataList1.DataBind()
        resultarray.Clear()
        Session("reportresult") = resultarray
      End If
    End Sub
```

```
  Private Sub exitbutton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles exitbutton.Click
    If reporttitletext.Text = "" Then
      reporttitle = "Final Report"
    Else
      reporttitle = reporttitletext.Text
    End If
    Session("reporttitle") = reporttitle
    Response.Redirect("continue.aspx")
  End Sub

  Private Sub Delete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Delete.Click
    Dim tablechoose
    Dim columnchoose
    tablechoose = Session("tablechoose")
    tempdataset = Session("tempdataset")

    If Not Session("reportresult") Is Nothing Then
      resultarray = Session("reportresult")
    End If
    If resultarray.Count = 0 Then
      Label5.Text = "No Item need to be Delete"
      resultarray.Add("It is Empty NOW!!!")
      DataList1.DataSource = resultarray
      DataList1.DataBind()
      resultarray.Clear()
    Else
      j = resultarray.Count
      Dim temptable As DataTable = tempdataset.Tables(tablechoose)
      columnchoose = resultarray(j - 1)
      Dim n = columnchoose.ToString.IndexOf("Record")
      If n > 0 Then
        tempdataset.Tables.Remove(tablechoose)
        If Not tempdataset.Tables.Count = 0 Then
          tablechoose = tempdataset.Tables(tempdataset.Tables.Count - 1).TableName
          Session("tablechoose") = tablechoose
        End If
      Else
        temptable.Columns.Remove(columnchoose)
      End If
      resultarray.RemoveAt(j - 1)
      DataList1.DataSource = resultarray
      DataList1.DataBind()
      Session("reportresult") = resultarray
      tempdataset.WriteXml(Server.MapPath("tempreport.xml"), XmlWriteMode.WriteSchema)
    End If
  End Sub

  Private Sub ReportSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ReportSearch.Click
    Dim datachoose2, columnchoose, tablechoose
    Dim searchdataset As New DataSet()
    datachoose2 = Session("datachoose2")
    columnchoose = Session("columnchoose")
    tablechoose = Session("tablechoose")
    Dim objCat
    Dim objTables
    Dim objTable
    Dim objColumn
```

```
        objCat = Server.CreateObject("ADOX.Catalog")
        objCat.ActiveConnection = "provider=microsoft.jet.oledb.4.0;" & "data source=" &
Server.MapPath(datachoose2)
        objTables = objCat.Tables
        'For Each objTable In objTables
        Dim connstr As String = "provider=microsoft.jet.oledb.4.0;data source=" &
Server.MapPath(datachoose2)
        Dim conn As OleDbConnection, adap As OleDbDataAdapter
        conn = New OleDbConnection(connstr)
        conn.Open()
        Dim sql
        'For Each objColumn In objTable.columns
        If Searchbox2.Text.ToString.Substring(0, 1) = "*" Then
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '%" &
Searchbox2.Text.ToString.Substring(1, Searchbox2.Text.ToString.Length - 1) & "'"
        ElseIf Searchbox2.Text.ToString.Substring(Searchbox2.Text.ToString.Length - 1, 1) = "*" Then
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '" &
Searchbox2.Text.ToString.Substring(0, Searchbox2.Text.ToString.Length - 1) & "%'"
        Else
            sql = "select * from " & tablechoose & " WHERE " & columnchoose & " LIKE '%" &
Searchbox2.Text & "%'"
        End If
        'Next
        adap = New OleDbDataAdapter(sql, conn)
        adap.Fill(searchdataset, tablechoose)
        searchdataset.WriteXml(Server.MapPath("searchsummary.xml"), XmlWriteMode.WriteSchema)
        conn.Close()
        'Next
        If searchdataset.Tables(0).Rows.Count = 0 Then
            Label5.Text = "Sorry, There is no result for your search."
        Else
            DataGrid2.DataSource = searchdataset
            DataGrid2.DataBind()
            DataList1.DataSource = resultarray
            DataList1.DataBind()
        End If
    End Sub

    Private Sub checkbutton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles checkbutton.Click
        Response.Redirect("checkdatabase.aspx")
    End Sub
End Class
```

Continue.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="continue.aspx.vb"
Inherits="WebApplication1.WebForm4"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>WebForm4</title>
        <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.0">
        <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
        <meta name="vs_defaultClientScript" content="JavaScript">
        <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    </HEAD>
    <body MS_POSITIONING="GridLayout">
        <form id="Form1" method="post" runat="server">
```

```
        <h1><asp:Label id="Label2" style="Z-INDEX: 105; LEFT: 38px; POSITION: absolute; TOP:
19px" runat="server" Width="375px" Height="19px" ForeColor="#004000" BorderStyle="None">Dynamic
Report Generator</asp:Label></h1>
        <H3>
            <asp:Label id="Label1" style="Z-INDEX: 101; LEFT: 90px; POSITION: absolute; TOP:
88px" runat="server">Do you want to continue?</asp:Label></H3>
        <asp:Button id="confirm" style="Z-INDEX: 102; LEFT: 91px; POSITION: absolute; TOP:
289px" runat="server" Text="Confirm"></asp:Button>
        <asp:RadioButtonList id="continue" style="Z-INDEX: 104; LEFT: 90px; POSITION: absolute;
TOP: 140px" runat="server" Width="307px" Height="117px" Font-Bold="True">
            <asp:ListItem Value="Back to Database Choose Page">Back to Database Choose
Page</asp:ListItem>
            <asp:ListItem Value="Back to Summary Page">Back to Summary Page</asp:ListItem>
            <asp:ListItem Value="Back to Report Page">Back to Report Page</asp:ListItem>
            <asp:ListItem Value="Exit" Selected="True">Exit</asp:ListItem>
        </asp:RadioButtonList>
    </form>
  </body>
</HTML>
```

continue.aspx.vb

```
Public Class WebForm4
  Inherits System.Web.UI.Page
  Protected WithEvents Label2 As System.Web.UI.WebControls.Label
  Protected WithEvents Label1 As System.Web.UI.WebControls.Label
  Protected WithEvents confirm As System.Web.UI.WebControls.Button
  Protected WithEvents continue As System.Web.UI.WebControls.RadioButtonList

#Region
  <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
  End Sub
  Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
    InitializeComponent()
  End Sub
#End Region

  Dim choose2
  Private Sub confirm_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
confirm.Click
    If continue.SelectedItem.Text = "Back to Database Choose Page" Then
      Server.Transfer("Choice2.aspx")
    ElseIf continue.SelectedItem.Text = "Back to Summary Page" Then
      Server.Transfer("Summary.aspx")
    ElseIf continue.SelectedItem.Text = "Back to Report Page" Then
      Server.Transfer("Report.aspx")
    ElseIf continue.SelectedItem.Text = "Exit" Then
      Server.Transfer("Close.aspx")
    End If
  End Sub
End Class
```

Close.aspx

```
<%@ Register TagPrefix="uc1" TagName="WebUserControl2" Src="WebUserControl2.ascx" %>
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="Close.aspx.vb"
Inherits="WebApplication1.WebForm5" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
```

```
    <HEAD>
        <title>WebForm5</title>
        <META content="Microsoft Visual Studio .NET 7.0" name="GENERATOR">
        <META content="Visual Basic 7.0" name="CODE_LANGUAGE">
        <META content="JavaScript" name="vs_defaultClientScript">
        <META content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    </HEAD>
    <BODY MS_POSITIONING="GridLayout">
        <FORM id="Form1" method="post" runat="server">
            <FONT face="新細明體">
                <asp:button id="sortbtn" style="Z-INDEX: 104; LEFT: 168px; POSITION: absolute; TOP:
198px" runat="server" Text="Sort" Width="59px"></asp:button><asp:dropdownlist
id="columnnamedatalist" style="Z-INDEX: 101; LEFT: 272px; POSITION: absolute; TOP: 199px"
runat="server" Width="112px"></asp:dropdownlist><asp:label id="Label3" style="Z-INDEX: 102; LEFT:
242px; POSITION: absolute; TOP: 203px" runat="server">By</asp:label><asp:dropdownlist
id="tablenamedatalist" style="Z-INDEX: 103; LEFT: 14px; POSITION: absolute; TOP: 199px"
runat="server" Width="149px" OnSelectedIndexChanged="Index_Changed"
AutoPostBack="True"></asp:dropdownlist><asp:datagrid id="DataGrid1" style="Z-INDEX: 105; LEFT:
18px; POSITION: absolute; TOP: 242px" runat="server" Width="244px" PageSize="25"
OnPageIndexChanged="changepage" HorizontalAlign="Left" AllowPaging="True" CellPadding="4"
BackColor="White" BorderWidth="1px" BorderStyle="None" BorderColor="#CC9966">
                    <SelectedItemStyle Font-Bold="True" ForeColor="#663399"
BackColor="#FFCC66"></SelectedItemStyle>
                    <ItemStyle ForeColor="#330099" BackColor="White"></ItemStyle>
                    <HeaderStyle Font-Bold="True" ForeColor="#FFFFCC"
BackColor="#990000"></HeaderStyle>
                    <FooterStyle ForeColor="#330099" BackColor="#FFFFCC"></FooterStyle>
                    <PagerStyle HorizontalAlign="Right" ForeColor="#330099" BackColor="#FFFFCC"
Mode="NumericPages"></PagerStyle>
                </asp:datagrid><asp:label id="label" style="Z-INDEX: 106; LEFT: 19px; POSITION:
absolute; TOP: 173px" runat="server" Font-Bold="True" ForeColor="Black">Final
Report</asp:label><uc1:webusercontrol2 id="ucsummary" runat="server"></uc1:webusercontrol2>
                <asp:Button id="Close" style="Z-INDEX: 107; LEFT: 411px; POSITION: absolute; TOP:
198px" runat="server" Text="Close the Window" Width="125px"></asp:Button></FONT></FORM>
    </BODY>
</HTML>
```

Close.aspx.vb

```
Imports System
Imports System.IO
Imports System.Xml
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.HtmlControls.HtmlGenericControl
Imports System.Web.UI.HtmlControls

Public Class WebForm5
    Inherits System.Web.UI.Page

#Region " Web Form 設計工具產生的程式碼 "

    '此呼叫為 Web Form 設計工具的必要項。
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
```

```
        'CODEGEN: 此方法呼叫為 Web Form 設計工具的必要項
        '請勿使用程式碼編輯器來修改它。
        InitializeComponent()
    End Sub

#End Region

    Dim i = 0
    Dim j = 0
    Protected ucsummary As WebUserControl2
    Protected WithEvents tablenamedatalist As System.Web.UI.WebControls.DropDownList
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents columnnamedatalist As System.Web.UI.WebControls.DropDownList
    Protected WithEvents sortbtn As System.Web.UI.WebControls.Button
    Protected WithEvents DataGrid1 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents label As System.Web.UI.WebControls.Label
    Protected WithEvents Close As System.Web.UI.WebControls.Button

    Dim sorttype
    Private currentDir As System.IO.Directory

    Private Sub page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        ucsummary.summaryreport = Session("summaryresult1")
        ucsummary.summarytitle = Session("summarytitle")

        Dim reporttitle
        If Not Page.IsPostBack Then
            Dim tablenamelist As New ArrayList()
            If Not Session("tablenamelist") Is Nothing Then
                tablenamelist = Session("tablenamelist")
                reporttitle = Session("reporttitle")
                label.Text = reporttitle
                tablenamedatalist.DataSource = tablenamelist
                tablenamedatalist.DataBind()
                Dim tablechoose = tablenamedatalist.SelectedItem.Text
                Dim columnnamelist As New ArrayList()
                Dim xmldoc As New XmlDataDocument()
                xmldoc.DataSet.ReadXml(("c:\dygtemp\" + "tempreport.xml"))
                For i = 0 To xmldoc.DataSet.Tables.Count - 1
                    If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
                        For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                            columnnamelist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
                        Next
                    End If
                Next i
                columnnamedatalist.DataSource = columnnamelist
                columnnamedatalist.DataBind()
                DataGrid1.DataSource = xmldoc.DataSet.Tables(tablechoose)
                DataGrid1.DataBind()
            End If
        End If

    End Sub

    Sub Index_Changed(ByVal sender As Object, ByVal e As EventArgs)
        Dim columnnamelist As New ArrayList()
        Dim tablechoose = tablenamedatalist.SelectedItem.Text
```

```
        Dim xmldoc As New XmlDataDocument()
        xmldoc.DataSet.ReadXml(("c:\dygtemp\" + "tempreport.xml"))
        For i = 0 To xmldoc.DataSet.Tables.Count - 1
            If tablechoose = xmldoc.DataSet.Tables(i).TableName.ToString Then
                For j = 0 To xmldoc.DataSet.Tables(i).columns.count - 1
                    columnnamelist.Add(xmldoc.DataSet.Tables(i).columns(j).columnname)
                Next
            End If
        Next i
        columnnamedatalist.DataSource = columnnamelist
        columnnamedatalist.DataBind()
        DataGrid1.DataSource = xmldoc.DataSet.Tables(tablechoose)
        DataGrid1.DataBind()
    End Sub

    Private Sub sortbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
sortbtn.Click
        Dim tablechoose
        Dim columnchoose
        sorttype = Session("sorttype")
        If sorttype = "" Then
            sorttype = " Desc"
        Else
            sorttype = ""
        End If
        tablechoose = tablenamedatalist.SelectedItem.Text
        columnchoose = columnnamedatalist.SelectedItem.Text
        Dim mydataset As New DataSet()
        mydataset.ReadXml(("c:\dygtemp\" + "tempreport.xml"))
        Dim mydatatable As New DataTable()
        mydatatable = mydataset.Tables(tablechoose.ToString)
        mydatatable.DefaultView.Sort = columnchoose & sorttype
        DataGrid1.DataSource = mydatatable.DefaultView
        DataGrid1.DataBind()
        Session("sorttype") = sorttype
    End Sub

    Sub changepage(ByVal sender As Object, ByVal e As DataGridPageChangedEventArgs)
        DataGrid1.CurrentPageIndex = e.NewPageIndex
        Dim tablechoose = tablenamedatalist.SelectedItem.Text
        Dim xmldoc As New XmlDataDocument()
        xmldoc.DataSet.ReadXml(("c:\dygtemp\" + "tempreport.xml"))
        DataGrid1.DataSource = xmldoc.DataSet.Tables(tablechoose)
        DataGrid1.DataBind()
    End Sub

    Private Sub Close_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Close.Click
        Dim f As New System.IO.FileInfo(("c:\dygtemp\"))
        Dim dir As New DirectoryInfo(("c:\dygtemp\"))
        For Each f In Dir.GetFiles("*.*")
            f.Delete()
        Next
        Response.Write("<script>window.returnValue = 'ok'; window.close();</script>")

    End Sub
End Class
```

WebUserControl2.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="WebUserControl2.ascx.vb"
Inherits="WebApplication1.WebUserControl2"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<P><asp:label id="Label1" runat="server" ForeColor="Black" Font-Bold="True">Label</asp:label></P>
<P><asp:datalist id="DataList1" BorderWidth="1px" GridLines="Both" CellPadding="4"
BackColor="White" BorderStyle="None" BorderColor="#CC9966" RepeatColumns="5" runat="server">
        <SelectedItemStyle Font-Bold="True" ForeColor="#663399"
BackColor="#FFCC66"></SelectedItemStyle>
        <ItemStyle ForeColor="#330099" BackColor="White"></ItemStyle>
        <ItemTemplate>
            <%# container.dataitem%>
        </ItemTemplate>
        <FooterStyle ForeColor="#330099" BackColor="#FFFFCC"></FooterStyle>
        <HeaderStyle Font-Bold="True" ForeColor="#FFFFCC" BackColor="#990000"></HeaderStyle>
    </asp:datalist></P>

WebUserCotrol2.ascx.vb
Public MustInherit Class WebUserControl2
    Inherits System.Web.UI.UserControl
    Protected WithEvents DataList1 As System.Web.UI.WebControls.DataList

#Region
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        InitializeComponent()
    End Sub
#End Region
    Public summaryreport As New ArrayList()
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Public summarytitle

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        If Not Session("summaryresult1") Is Nothing Then
            summaryreport = Session("summaryresult1")
            DataList1.DataSource = summaryreport
            DataList1.DataBind()
            summarytitle = Session("summarytitle")
            Label1.Text = summarytitle
        End If
    End Sub
End Class
```

APPENDIX B

SOFTWARE REQUIREMENT SPECIFICATION

B. SRS
Master's Project
Software Requirement Specification

Dynamic Report Generator

Advisor: Dr. Kay Zemoudeah
Committee: Dr. David Turner
Dr. Kerstin Voigt
Student: Li-Chuan Hsieh
E-Mail: jack1201@yahoo.com
Date: January 13, 2003

Table of Content

1.0 Introduction

1.1 Purpose of this Document

This Software Requirements Specification (SRS) document provides a description of the purpose, functionality, and interface of the proposed Dynamic Report Generator Program (DRG). This program will provide a drag-and-drop non-programming user interface. Users can generate a report from more than one database or XML documents. The report that user generates is based on user GUI commands. User requirements for the program are based on Java language, Java bean, JSP, Visual Basic, Mysql, XML, SHTTP, and SSL.

1.2 Scope of the Development Project

The DRG will provide a quick, safe and convenient way for internet users to generate a report from a DataBase or XML document. The program will be java based. The user will be able to drag and drop labels, "functional boxes," and list of items. This program will be very similar to VB.net. It is designed to service clients anywhere in the world. This software product has three main functions. The user gives the request and command on a web page. After receiving a command, the DRG will be activated. The DRG program will start getting data and report from database or XML document. Before generating or sorting the data, the DRG would convert DBs to XML documents. When all needed data have been collected, the DRG program will begin to generate the data and build a new format structure. All data and generated results will be sorted based on the user's requests and they will also be put in a report document.(3) When the new report has been finished, the DRG program will send back the new report to the user's homepage.

1.3 Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specifications.

DRG – Dynamic Report Generator.

HTML – Hyper Text Markup Language.

Visual Basic – Mircosoft Visual Basic Program.

Java – An object oriented language developed by Sun Microsystems, Java programs are capable of running on most popular computer platforms without the need for recompilation.

JDBC – (Java Database Connectivity) a programming interface that lets Java applications access a database via the SQL language.

MYSQL – MYSQL is a software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

XML - Extensible Markup Language, XML is subset of the Standard Generalized Markup Language (SGML) defined in ISO standard 8879:1986 that is designed to make it easy to interchange structured documents over the Internet. XML files always clearly mark where the start and end of each of the logical parts (called *elements)* of an interchanged document occurs. XML restricts the use of SGML constructs to ensure that fall back options are available when access to certain components of the document are not currently possible over the Internet. It also defines how Internet Uniform Resource Locators can be used to identify component parts of XML data streams.

JSP – Java Server Page, an extension of the Java server technology from Sun that provides a simple programming vehicle for displaying dynamic content on a Web page.

Java Servlet – A Java application that runs in a Web server or application server and provides server-side processing, typically to access a database or perform e-commerce processing.

JavaScript – A scripting language that is widely supported in Web browser and other Web tools. It adds interactive functions to HTML pages, which are otherwise static.

1.4 References

[1] IEEE Std. 830-1998 IEEE Recommended Practice of Software Requirements Specifications – Annex A

[2] xlinkit: A Consistency Checking and Smart Link Generation Service, Nentwich C., Capra L., Emmerich W., Finkelstein A., ACM Transactions on Internet Technology, Vol. 2, No. 2, p-p 151-185, May 2002.

[3] Naked objects: a technique for designing more expressive systems, Pawson R., Matthews R., ACM SIGPLAN Notices, V. 36(12), December 2001

[4] O'REILLY XML.COM, http://www.xml.com/

[5] The Java Language Specification, http://java.sun.com/docs/books/jls/

[6] MySQL Homepage, http://www.mysql.com/

[7] JDBC Data Access API, http://java.sun.com/products/jdbc/

[8] Java™ Servlet Technology, http://java.sun.com/products/servlet/

1.5 Overview of Document

Section two of this document follows the guidelines of IEEE Std. 830-1998 IEEE Recommended Practice of Software Requirements Specifications. This section provides product perspective, a summary of DRG's functions, a description of the characteristics of the expected users, and a list of assumptions and dependencies. Section three of this document presents the specific requirements for this system. They are organized by mode, following the SRS Section three template shown in IEEE Std 830-1998, Annex A, and Paragraph A.3.

2.0 Overall Description

2.1 Project perspective

The DRG project's purpose is to make a flexible non-programming report generator. The DRG program will run in the user's homepage and the data will transfer between the database and the DRG program. Before the data go into the program, the DBs data would convert to XML document. Generating and sorting out of data will be done in the DRG program. The interfaces of this program must support current versions of Netscape & Internet Explorer, which can execute JDBC commands. The communication interfaces require support for Secure Hyper Text Transfer Protocol (SHTTP) and Secure Sockets layer (SSL).

2.1.1 System Interfaces

The DRG program is a 3-tier distributed architecture. The first tier displays the user interfaces in a Web browser using HTML. In this tier, the user sends request and command to the main program. The middle tier is the main program of DRG. In this tier, the DRG gets the data from databases, converts data to XML document and sorts out the data and results in a new format. Then send back the new report to user's interface. The third tier is the database. In this tier, the DRG main program would collect the data from the database or XML document. (Figure 1).

2.1.2 User Interfaces

A user can check the original data and new report from a web browser which has a DRG program.

2.1.3 Hardware

The DRG needs the following hardware.

(1) PC with a 56k+ Internet connection.

(2) 1GHZ Pentium or faster.

(3) 128MB RAM or more.

For the server and client computers, the hardware requirements are:

(1) PC with a 56k+ Internet connection.

(2) 1GHZ Pentium or faster.

(3) 128MB RAM or more.

2.1.4 Software Interfaces

Software interfaces are provided in Java 1.2 APIs or higher, JSP APIs, SSL, JDBC APIs, MYSQL, XML and SSH. In addition, client computers need a compatible web browser (Netscape 4.0 or higher, Internet Explorer 4.0 or higher). The web browser needs to implement JSP and SSL and provides SHTTP services. The user needs SSH software to connect to server. A JDBC driver is needed for the database. Alternatively MYSQL might be used.

2.1.5 Communications Interfaces

The communications between the DRG program and the database is through SSL. All information provided to users is through SHTTP (Figure 2).

2.1.6 Memory Constraints

Enough memory is required to guarantee acceptable response time. 128 Mega bytes or higher is recommended.

2.1.7 Operations

DRG will be accessed through the SHTTP. It will be initiated and terminated by Internet users. The user's request will be sent to the DRG program. When the DRG program receives the request, it will start by getting the data from the database and convert the DB data to XML document before the report is generated. Inside the DRG program, the data will be generated and sorted out by the user's order. At the same time, the DRG program will also structure a new style sheet to follow the user's

request. When all work is done, the results will be sent back to the user's interface. The DRG also can directly operate on XML.

2.2 Product functions

Figure 2 shows the User DRG/Database diagram that graphically depicts the client and principal functions of DRG. The functions will be further described in subsections 2.2.1 and 2.2.2 and the actors in the diagram are further described in section.

2.2.1 Use SSL and SHTTP

Between DRG and the user, the connection is STHHP, which can transfer encrypted information and JDBC commands from user to the DRG program. The connection between the DRG and the database is using the SSL.

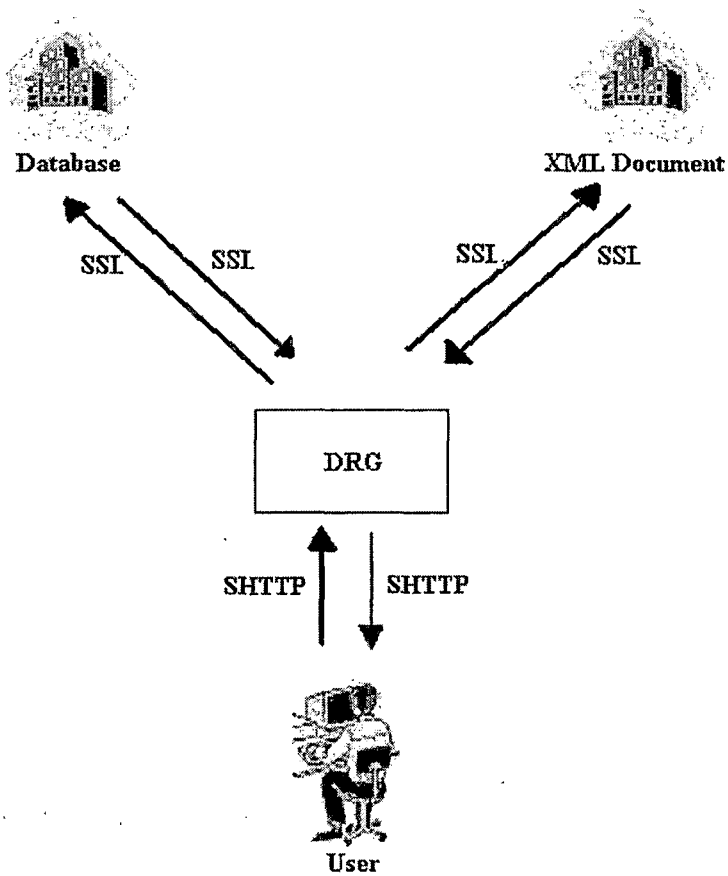

Figure 2. Communication Diagram

2.2.2 Encryption and Decryption

Before transferring data between the DRG program and databases, the data have to be encrypted with a code for security reasons. After the receiving side gets the data, the DRG program should be able to decrypt the data.

2.2.3 Convert DBs to XML document

Before generating the report, the DRG will convert the data from the database to an XML document.

2.3 User Characteristics

Users who use DRG program are expected to know how to use a web browser, SSH and speak English. They should also be able to follow the manual written in plain English.

2.4 Assumption and Dependencies

These requirements assume there are no applicable hardware limitations. It is also assumed that system administration and maintenance will be done with by the host system or system administrator.

3.0 Specific requirements

This section contains the software requirements to a level of sophistication that would enable us to design the DRG in conformance with the requirements of this Specification Requirement Specification document. This level of sophistication will also enable testers to generate tests for the system and to verify whether it meets the requirements. Every stated requirement will be externally perceivable by users through the usage of sample screen dumps. The requirements include a description of every input, every output, and all functions performed by the system to generate output in response to input.

3.1 External interface requirements

This is a detailed description of the inputs, outputs and concepts of the DRG.

3.1.1 Page of Connection between User and DRG program

In this page, the tester can use the hyperlink from the main page to detect the status of the Internet connection between the user and the DRG program.

3.1.2 Page of connection between DRG program and Database

In this page, the tester can use the hyperlink from main page to detect the status of Internet connection between the DRG program and the database.

3.1.3 Page for sending command to DRG program

This page allows the tester to send his/her command to the DRG program to upload the data from the database. Then the DRG could generate all information, build a new structure for all data, and sort out data in the DRG program.

3.1.4 Page for User receiving the New Report

Users should be able to receive the new report that includes data and results, which are displayed immediately in a new interface with a new structure.

3.2 Performance requirements

The DRG is an internet interface program, so it will support all users who through the DRG program can receive data on the internet. This is done quickly and conveniently. The response time to view any page should be less than five seconds. The response time should not exceed twice the download length when viewing a page.

3.3 Software system attributes

3.3.1 Security

Security is one of the biggest issues in this project. DRG must make sure that intruders would not get the data from the database. On the other hand, in the communication system, we need to know the Internet status on the connection between server, database and client. The security systems of DRG program are focused in SSL and SHTTP. SSL is between database and DRG program, and SHTTP is between DRG program and user's interface.

3.3.1.1SSL

Secure Socket Layer (SSL) can authenticate the identity of communicating terminals to the server. SSL use the symmetric key for the session communication. Symmetric keys are used for encryption because of their computational simplicity relative to public key cryptography. The system manages a cache of available connections to remotes servers to avoid the inconvenience of establishing a secure connection for each logical operation.

When terminals send request to the server, all communications should use SSL. If the communication fails, terminals should re-send the request. We can see the communication between the server and the terminals in Figure 3.

The steps of communication based on the figure3 are:

1) The terminal sends a message to the server to make a connection.

2) The server sends the terminal a response that the connection is established.

3) The terminal sends the request asking for the data.

4) The server returns the data and transfer status.

5) Continue step 3 and step 4 until the data file is completely sent to the client.

6) The server terminates the connection.

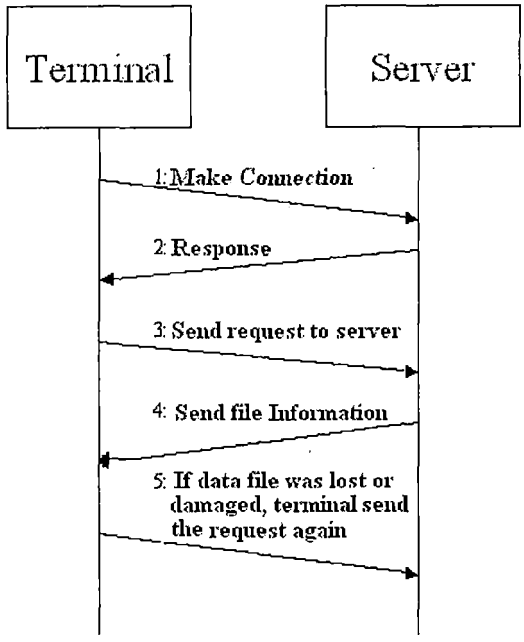

Figure 3 SSL Communication Diagram

3.3.2 Mysql Database

Database is a very important part of this project. DRG program will operate on XML document, and translates DB data to XML before acting on it. In this project, the Mysql has been chosen to be the database.

MySQL, the most popular Open Source SQL database, is developed, distributed and supported by MySQL AB. MySQL AB is a commercial company founded by the MySQL developers that builds its business providing services around the MySQL database.

MySQL is a database management system.A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management plays a central role in computing, as stand-alone utilities, or as parts of other applications.

MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The tables are linked by defined relations making it possible to combine data from several tables on request. The SQL part of ``MySQL'' stands for "Structured Query Language"—the most common standardized language used to access databases.

MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify. Anybody can download the MySQL software from the Internet and use it without paying anything. Anybody so inclined can study the source code and change it to fit their needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application you can buy a commercially licensed version from us.

At the final, because the MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of programming interfaces (APIs). So the MySQL Server can be a multi-threaded library which everyone can link into any application to get a smaller, faster, easier-to-manage product.

3.3.3 XML Document

XML, Extensible Markup Language, is the universal format for structured documents and data on the web and is certificated by W3C. It is a language similar to HTML, but it is more powerful and flexible. Using XML, we can define our own tags whose names suggest the meaning of its content, and furthermore, XML is license-free, platform-independent well supported.

By defining the role of each element of text in a formal model, known as a *Document Type Definition* (DTD), users of XML can check that each component of document occurs in a valid place within the interchanged data stream. An XML DTD allows computers to check, for example, that users do not accidentally enter a third-level heading without first having entered a second-level heading, something that cannot be checked using the HyperText Markup Language (HTML) previously used to code documents that form part of the World Wide Web (WWW) of documents accessible through the Internet.

However, unlike SGML, XML does not require the presence of a DTD. If no DTD is available, either because all or part of it is not accessible over the Internet or because the user failed to create it, an XML system can assign a default definition for undeclared components of the markup.

XML allows users to:

- bring multiple files together to form compound documents
- identify where illustrations are to be incorporated into text files, and the format used to encode each illustration
- provide processing control information to supporting programs, such as document valuators and browsers
- Add editorial comments to a file.

It is important to note, however, that XML is not:

- a predefined set of tags, of the type defined for HTML, that can be used to markup documents
- A standardized template for producing particular types of documents.

XML was not designed to be a standardized way of coding text: in fact it is impossible to devise a single coding scheme that would be suit all languages and all applications. Instead XML is formal language that can be used to pass information about the component parts of a document to another computer system. XML is flexible enough to be able to describe any logical text structure, whether it be a form, memo, letter, report, book, encyclopedia, dictionary or database.

3.3.4 Maintainability

The DRG system consists of Java language, Java beans, Java classes, SSL, JSP pages and style sheets. They will be put under different directories with hierarchy.

Other files and documents also will be put in separate directories as well. This structure will aid in maintaining all modules organized and therefore maximizing maintenance facility.

3.3.5 Portability

The Server-Implementation of the DRG should be 100% portable since it will be written in java, it is proven portable languages. The only determinant of how easily the DRG is ported from on architecture to another is having the latest version of the Java Virtual Machine installed on the web-server machine.

The terminal portion of the DRG wills also 100% portable since the system will be in presented using dynamic HTML pages and style sheet, which is supported by most up-to-date web browser.

3.3.6 Reliability

All contents and logs shall be generated dynamically and automatically so no human interference is needed, unless a programmer who want to fix the security function. The server shall be twenty-four hours a day and seven days a week, with exception that periodical system maintaining needs to be conducted depending on the reliability of the web server. At least, the system should handle network packet loss smoothly.

REFERENCES

[1] Developer Forums: Java Programming
http://forum.java.sun.com/forum.jsp?forum=31

[2] Borland USA
http://www.borland.com/products/downloads/
download_jbuilder.html

[3] O'REILLY XML.COM
http://www.xml.com/

[4] Microsoft Asp.Net
http://www.asp.net/Default.aspx?tabindex=0&tabid=1

[5] Microsoft Visual Studio Developer Center,
http://msdn.microsoft.com/vstudio/

[6] Apache HTTP Server Version 1.3, Security Tips for
Server Configuration http://httpd.apache.org/docs/
misc/security_tips.html

[7] L. D. Stein, J. N. Stewart, The World Wide Web
Security FAQ, Version 3.1.2, February 4, 2002
http://www.w3.org/Security/Faq/www-security-faq.html

[8] D. Flanagan, JAVA IN A NUTSHELL, 2nd Edition, May
1997

[9] Pearson Education, Inc., JAVA, HOW TO PROGRAM, Fifth
Edition, 2003

[10] Hallogram Publishing, JavaBean Dynamic Report
Generator http://www.hallogram.com/jviewpro/

[11] The Big Faceless Organization, The Big Faceless
Report Generator http://big.faceless.org/
products/report/

[12] NOVAtime Technology, Inc., Dynamic Grouping Report
Generator http://www.novatime.com/
product_features_reportgenerator.htm

[13] Report Generator http://www.apdl.co.uk/isv/repgen.htm

[14] B. Rimmington, User Report Generator (URG)
http://www.dbase.com/Knowledgebase/
dbulletin/bu09urg.htm